



Guida per l'utente per la versione 1

# AWS Command Line Interface



# AWS Command Line Interface: Guida per l'utente per la versione 1

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

---



# Table of Contents

.....	xiv
Informazioni su AWS CLI .....	1
Informazioni sulla versione 1 AWS CLI .....	2
Manutenzione e supporto per le versioni principali SDK .....	2
Informazioni su Amazon Web Services .....	2
Informazioni sugli esempi .....	3
Documentazione e risorse aggiuntive .....	4
AWS CLI documentazione e risorse .....	4
Altro e strumenti AWS SDKs .....	4
Installare il AWS CLI .....	6
Requisiti della versione di Python .....	6
Amazon Linux .....	7
Prerequisiti .....	7
pip .....	8
yum .....	9
Risoluzione degli errori di AWS CLI installazione e disinstallazione .....	10
Linux .....	10
Prerequisiti .....	11
Installazione e disinstallazione utilizzando il programma di installazione in bundle .....	11
Installazione e disinstallazione utilizzando pip .....	17
Installazione e disinstallazione utilizzando pip .....	20
Aggiungi l'eseguibile della AWS CLI versione 1 al percorso della riga di comando .....	22
Risoluzione degli errori di AWS CLI installazione e disinstallazione .....	23
macOS .....	24
Prerequisiti .....	24
Installazione e disinstallazione utilizzando il programma di installazione in bundle .....	25
Installazione e aggiornamento utilizzando pip .....	30
Risoluzione degli errori di AWS CLI installazione e disinstallazione .....	34
Windows .....	34
Installa, aggiorna e disinstalla utilizzando il programma di MSI installazione .....	35
Installazione, aggiornamento e disinstallazione mediante Python e pip su Windows .....	37
Aggiungi l' AWS CLI eseguibile al percorso della riga di comando .....	38
Risoluzione degli errori di AWS CLI installazione e disinstallazione .....	40
Virtualenv .....	40

Prerequisiti .....	41
Installazione e aggiornamento in un ambiente virtuale .....	41
Risoluzione degli errori di AWS CLI installazione e disinstallazione .....	42
Configurare il AWS CLI .....	43
Configurazione e precedenza delle credenziali .....	43
Argomenti aggiuntivi in questa sezione .....	44
Impostazioni dei file di configurazione e credenziali in AWS CLI .....	44
Formato dei file di configurazione e delle credenziali .....	45
Dove vengono archiviate le impostazioni di configurazione? .....	50
Utilizzo di profili denominati .....	51
Imposta e visualizza le impostazioni di configurazione utilizzando i comandi .....	52
Impostazione di nuovi esempi di comandi per la configurazione e le credenziali .....	54
Impostazioni supportate del file config .....	55
Variabili di ambiente .....	73
Come impostare le variabili di ambiente .....	74
AWS CLI variabili di ambiente supportate .....	75
Opzioni della riga di comando in AWS CLI .....	85
Come utilizzare le opzioni della riga di comando .....	86
AWS CLI opzioni della riga di comando globali supportate .....	86
Usi comuni delle opzioni della riga di comando .....	90
Configurazione del completamento dei comandi in AWS CLI .....	90
Come funziona .....	91
Configurazione del completamento dei comandi su Linux o macOS .....	92
Configurazione del completamento dei comandi in Windows .....	95
Tentativi .....	96
Modalità di riprova disponibili .....	97
Configurazione di una modalità di ripetizione .....	100
Visualizzazione dei registri dei tentativi di nuovo tentativo .....	101
Utilizzo di un HTTP proxy per AWS CLI .....	101
Utilizzo degli esempi .....	102
Autenticazione in un proxy .....	103
Utilizzo di un proxy su EC2 istanze Amazon .....	104
Risoluzione dei problemi .....	104
Endpoints .....	104
Imposta l'endpoint per un singolo comando .....	105
Imposta un endpoint globale per tutti Servizi AWS .....	105

Imposta per utilizzare gli endpoint per tutti FIPs Servizi AWS .....	107
Imposta per utilizzare endpoint dual-stack per tutti Servizi AWS .....	108
Imposta endpoint specifici del servizio .....	109
Priorità della configurazione e delle impostazioni degli endpoint .....	112
Credenziali di autenticazione e accesso .....	114
Configurazione e precedenza delle credenziali .....	114
Argomenti aggiuntivi in questa sezione .....	115
Credenziali a breve termine .....	115
IAMruoli .....	117
Prerequisiti .....	117
Panoramica sull'utilizzo dei IAM ruoli .....	117
Configurazione e utilizzo di un ruolo .....	119
Usando MFA .....	121
Ruoli per più account e ID esterno .....	122
Specifica di un nome di sessione del ruolo per semplificare l'audit .....	123
Assunzione di un ruolo con un'identità Web .....	124
Cancellazione delle credenziali nella cache .....	125
IAMutenti .....	126
Fase 1: Crea il tuo IAM utente .....	126
Passaggio 2: ottieni le tue chiavi di accesso .....	126
Configura il AWS CLI .....	127
Utilizzo dei metadati delle EC2 istanze Amazon come credenziali nel AWS CLI .....	128
Prerequisiti .....	128
Configurazione di un profilo per i metadati Amazon EC2 .....	129
Credenziali esterne .....	130
Utilizzando il AWS CLI .....	133
Chiedere aiuto .....	134
Il comando AWS CLI help integrato .....	134
AWS CLI guida di riferimento .....	139
APIdocumentazione .....	139
Risoluzione degli errori .....	140
Aiuto aggiuntivo .....	140
Struttura di comando .....	140
Struttura dei comandi .....	140
Comandi Wait .....	141
Specificare i valori dei parametri .....	143

Tipi di parametri comuni .....	144
Citazioni con stringhe .....	149
Parametri dai file .....	153
Genera un modello di CLI scheletro .....	157
Sintassi abbreviata .....	163
Output del comando di controllo .....	165
Uscita sensibile .....	166
Opzioni di output lato server o lato client .....	166
Formato di output .....	167
Paginazione .....	174
Uscita del filtro .....	177
Codici restituiti .....	201
Alias .....	203
Prerequisiti .....	203
Fase 1: Creazione del file alias .....	203
Fase 2: Creazione di un alias .....	205
Fase 3: Chiamare un alias .....	208
Esempi di repository di alias .....	210
Risorse .....	211
Esempi di codice .....	213
Esempi di comandi guidati .....	213
DynamoDB .....	214
Amazon EC2 .....	218
S3 Glacier .....	237
IAM .....	244
Amazon S3 .....	248
Amazon SNS .....	267
Esempi di comando .....	270
ACM .....	276
APIGateway .....	287
APIGateway HTTP e WebSocket API .....	344
APIGestione del gateway API .....	390
App Mesh .....	392
App Runner .....	437
AWS AppConfig .....	472
Application Auto Scaling .....	506

Application Discovery Service .....	524
AppRegistry .....	530
Athena .....	541
Auto Scaling .....	575
Piani di Auto Scaling .....	645
AWS Backup .....	653
AWS Batch .....	659
Budget AWS .....	673
Amazon Chime .....	684
Cloud Control API .....	757
AWS Cloud Map .....	763
AWS Cloud9 .....	772
AWS CloudFormation .....	781
CloudFront .....	830
Amazon CloudSearch .....	900
CloudTrail .....	901
CloudWatch .....	918
CloudWatch Registri .....	932
CloudWatch Monitoraggio della rete .....	938
CodeArtifact .....	951
CodeBuild .....	979
CodeCommit .....	1044
CodeDeploy .....	1118
CodeGuru Revisore .....	1158
CodePipeline .....	1177
AWS CodeStar Notifiche .....	1209
CodeConnections .....	1220
Amazon Cognito Identity .....	1228
Provider di identità Amazon Cognito .....	1234
Amazon Comprehend .....	1302
Amazon Comprehend Medical .....	1438
AWS Config .....	1474
Amazon Connect .....	1498
AWS Cost and Usage Report .....	1515
Servizio Cost Explorer .....	1518
Firehose .....	1526

Amazon Data Lifecycle Manager .....	1529
AWS Data Pipeline .....	1535
DataSync .....	1545
DAX .....	1549
Test .....	1568
Device Farm .....	1580
AWS Direct Connect .....	1585
AWS Directory Service .....	1636
AWS DMS .....	1638
Amazon DocumentDB .....	1682
DynamoDB .....	1739
DynamoDB Streams .....	1835
Amazon EC2 .....	1842
Amazon EC2 Instance Connect .....	2507
Amazon ECR .....	2508
Amazon ECR pubblico .....	2539
Amazon ECS .....	2546
Amazon EFS .....	2631
Amazon EKS .....	2639
Elastic Beanstalk .....	2718
Elastic Load Balancing - Versione 1 .....	2748
Elastic Load Balancing - Versione 2 .....	2777
Elastic Transcoder .....	2831
ElastiCache .....	2858
MediaStore .....	2964
Amazon EMR .....	2981
Amazon EMR su EKS .....	3031
EventBridge .....	3032
Firewall Manager .....	3038
AWS FIS .....	3048
Amazon GameLift .....	3067
Global Accelerator .....	3100
AWS Glue .....	3139
GuardDuty .....	3161
AWS Health .....	3179
HealthImaging .....	3186

HealthLake .....	3214
HealthOmics .....	3225
IAM .....	3293
Access Analyzer di IAM .....	3430
Image Builder .....	3467
Strumento di gestione degli incidenti .....	3508
Contatti di Incident Manager .....	3530
Amazon Inspector .....	3554
AWS IoT .....	3598
AWS IoT 1-Click Dispositivi .....	3778
AWS IoT 1-Click Progetti .....	3788
AWS IoT Analytics .....	3800
Device Advisor .....	3826
AWS IoT data .....	3841
AWS IoT Events .....	3844
AWS IoT Events-Data .....	3869
AWS IoT Greengrass .....	3894
AWS IoT Greengrass V2 .....	3980
AWS IoT Jobs SDK release .....	4006
AWS IoT SiteWise .....	4009
AWS IoT Things Graph .....	4059
Wireless AWS IoT .....	4086
Amazon IVS .....	4123
IVSChat Amazon .....	4162
Streaming IVS in tempo reale di Amazon .....	4175
Amazon Kendra .....	4205
Kinesis .....	4214
AWS KMS .....	4233
Lake Formation .....	4303
Lambda .....	4354
License Manager .....	4396
Lightsail .....	4409
Macie .....	4535
Grafana gestito da Amazon .....	4540
MediaConnect .....	4542
MediaConvert .....	4558

MediaLive .....	4582
MediaPackage .....	4589
MediaPackage VOD .....	4604
MediaStore Piano dati .....	4616
MediaTailor .....	4622
MemoryDB .....	4627
Amazon MSK .....	4664
Network Manager .....	4673
Nimble Studio .....	4710
OpenSearch Servizio .....	4729
AWS OpsWorks .....	4743
AWS OpsWorks CM .....	4799
Organizations .....	4815
AWS Outposts .....	4853
AWS Payment Cryptography .....	4857
AWS Payment Cryptography Piano dati .....	4877
Amazon Pinpoint .....	4887
Amazon Polly .....	4910
Listino prezzi AWS .....	4916
AWS Private CA .....	4921
AWS Proton .....	4929
QLDB .....	4941
Amazon RDS .....	4964
Servizio RDS dati Amazon .....	5161
Amazon RDS Performance Insights .....	5165
Amazon Redshift .....	5169
Amazon Rekognition .....	5249
AWS RAM .....	5326
Resource Explorer .....	5350
Gruppi di risorse .....	5372
Etichettatura dei Resource Groups API .....	5385
AWS RoboMaker .....	5389
Route 53 .....	5425
Registrazione del dominio Route 53 .....	5439
Profili Route 53 .....	5466
Route 53 Resolver .....	5477



---

Amazon S3 .....	5522
Controllo Amazon S3 .....	5614
S3 Glacier .....	5630
Secrets Manager .....	5652
Security Hub .....	5681
Security Lake .....	5760
AWS Serverless Application Repository .....	5794
Service Catalog .....	5796
Service Quotas (Quote di Servizio) .....	5828
Amazon SES .....	5838
Scudo .....	5851
Signer .....	5867
Snowball .....	5877
Amazon SNS .....	5878
Amazon SQS .....	5900
Storage Gateway .....	5921
AWS STS .....	5924
AWS Support .....	5933
Amazon SWF .....	5946
Systems Manager .....	5962
Amazon Textract .....	6138
Amazon Transcribe .....	6149
Amazon Translate .....	6193
Trusted Advisor .....	6194
Autorizzazioni verificate .....	6214
VPCReticolo .....	6240
AWS WAF Classic .....	6268
AWS WAF Classic regionale .....	6273
AWS WAFV2 .....	6279
Amazon WorkDocs .....	6324
Amazon WorkMail .....	6357
Flusso di WorkMail messaggi Amazon .....	6381
WorkSpaces .....	6383
X-Ray .....	6398
esempi di script Bash .....	6416
DynamoDB .....	6417

Amazon EC2 .....	6489
HealthImaging .....	6595
IAM .....	6604
Amazon S3 .....	6659
AWS STS .....	6683
Sicurezza .....	6686
Protezione dei dati .....	6687
Crittografia dei dati .....	6688
Identity and Access Management .....	6688
Destinatari .....	6689
Autenticazione con identità .....	6689
Gestione dell'accesso con policy .....	6693
Come Servizi AWS lavorare con IAM .....	6695
Risoluzione dei problemi relativi AWS all'identità e all'accesso .....	6695
Convalida della conformità .....	6697
Resilienza .....	6699
Sicurezza dell'infrastruttura .....	6699
Applicazione di una versione minima TLS .....	6700
Risolvere gli errori .....	6704
Risoluzione dei problemi generali da provare prima .....	6704
Controlla la formattazione AWS CLI dei comandi .....	6705
Verifica che il Regione AWS tuo AWS CLI comando stia usando .....	6705
Conferma che stai utilizzando una versione recente di AWS CLI .....	6706
Usa l' --debugopzione .....	6706
Abilita e rivedi i registri della cronologia dei AWS CLI comandi .....	6712
Conferma che il tuo AWS CLI sia configurato .....	6712
Errori relativi al comando non trovato .....	6713
Il comando <code>aws --version ""</code> restituisce una versione diversa da quella installata .....	6716
Il comando <code>"aws --version"</code> restituisce una versione dopo la disinstallazione di AWS CLI .	6717
Ha AWS CLI elaborato un comando con un nome di parametro incompleto .....	6718
Errori di accesso negato .....	6719
Credenziali non valide ed errori chiave .....	6720
La firma non corrisponde agli errori .....	6722
Nessun errore rilevato dalla console Windows .....	6723
SSLerrori nei certificati .....	6724
Errori non validi JSON .....	6725

---

Risorse aggiuntive .....	6727
Cronologia dei documenti .....	6728

Questa documentazione è valida AWS CLI solo per la versione 1. Per la documentazione relativa alla versione 2 di AWS CLI, consulta la [Guida per l'utente della versione 2](#).

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.

# Cos'è la AWS Command Line Interface versione 1?

## Note

La AWS CLI versione 1 non è la versione più recente di AWS CLI. Alcune funzionalità introdotte nella AWS CLI versione 2 non sono state trasferite alla versione 1 ed è necessario eseguire l'aggiornamento per accedere a tali funzionalità. Esistono alcune modifiche «importanti» rispetto alla versione 1 che potrebbero richiedere la modifica degli script. Per un elenco delle modifiche più importanti nella versione 2, consulta [la Guida per l'utente della AWS CLI versione 2](#).

Il AWS Command Line Interface (AWS CLI) è uno strumento open source che consente di interagire con i AWS servizi utilizzando i comandi della shell della riga di comando. Con una configurazione minima, AWS CLI consente di iniziare a eseguire comandi che implementano funzionalità equivalenti a quelle fornite dal browser AWS Management Console dal prompt dei comandi del programma terminale:

- Shell Linux: utilizza programmi shell comuni come [bash](#), [zsh](#), e [tcsh](#) per eseguire comandi in Linux o macOS.
- Righe di comando di Windows: in Windows, esegui i comandi dal prompt dei comandi di Windows o in PowerShell.
- In remoto: esegui comandi su istanze Amazon Elastic Compute Cloud EC2 (Amazon) tramite un programma per terminali remoti come PuTTY o SSH, o con AWS Systems Manager.

Tutte le funzioni di AWS amministrazione, gestione e accesso IaaS (infrastructure as a service) AWS Management Console disponibili nell'AWS API e AWS CLI. Le nuove funzionalità e servizi AWS IaaS forniscono AWS Management Console funzionalità complete fino a CLI al momento del lancio API o entro 180 giorni dal lancio.

AWS CLI Fornisce l'accesso diretto alle API pubbliche dei servizi AWS. È possibile esplorare le funzionalità di un servizio e sviluppare script di shell per gestire le risorse. AWS CLI Oltre ai comandi API equivalenti a basso livello, diversi servizi AWS forniscono personalizzazioni per AWS CLI. Le personalizzazioni possono includere comandi di livello superiore che semplificano l'utilizzo di un servizio con un'API complessa.

## Informazioni sulla versione 1 AWS CLI

La AWS CLI versione 1 è l'originale AWS CLI e continuiamo a supportarla. Tuttavia, le nuove funzionalità principali introdotte nella AWS CLI versione 2 potrebbero non essere trasferite nella AWS CLI versione 1. Per utilizzare queste funzionalità, è necessario installare la AWS CLI versione 2. La AWS CLI versione 1 è stata creata utilizzando SDK for Python e pertanto richiede l'installazione di una versione compatibile di Python.

Per installare la AWS CLI versione 1, vedi [Installare il AWS CLI](#)

Per verificare la versione attualmente installata, utilizzare il seguente comando:

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

Per la cronologia delle versioni, consulta il [Changelog della AWS CLI versione 1](#) su GitHub

## Manutenzione e supporto per le versioni principali SDK

Per informazioni sulla manutenzione e il supporto per le versioni SDK principali e le relative dipendenze sottostanti, consulta quanto segue nella [Guida di riferimento agli strumenti AWS SDKs e agli strumenti](#):

- [AWS SDKse politica di manutenzione degli strumenti](#)
- [AWS SDKse matrice di supporto delle versioni degli strumenti](#)

## Informazioni su Amazon Web Services

Amazon Web Services (AWS) è una raccolta di servizi di infrastruttura digitale che gli sviluppatori possono sfruttare per sviluppare le loro applicazioni. I servizi includono elaborazione, archiviazione, database e sincronizzazione delle applicazioni (messaggistica e accodamento). AWS utilizza un modello di servizio pay-as-you-go . Ti vengono addebitati solo i servizi utilizzati dall'utente o dalle sue applicazioni. Inoltre, per renderla AWS più accessibile come piattaforma per la prototipazione e la sperimentazione, offre un livello di utilizzo gratuito. AWS In questo piano, i servizi sono gratuiti al di sotto di un determinato livello di utilizzo. [Per ulteriori informazioni sui AWS costi e sul piano gratuito, consulta Free Tier.AWS](#) Per creare un AWS account, apri la [AWS home page](#) e scegli Crea un AWS account.

## Informazioni sugli esempi nella Guida per l' AWS CLI utente

Gli esempi AWS Command Line Interface (AWS CLI) di questa guida sono formattati utilizzando le seguenti convenzioni:

- Prompt: il prompt dei comandi utilizza il prompt di Linux e viene visualizzato come `()`. `$` Per i comandi specifici di Windows, `C:\>` viene utilizzato come prompt. Non includere il prompt quando digiti i comandi.
- Directory — Quando i comandi devono essere eseguiti da una directory specifica, il nome della directory viene visualizzato prima del simbolo del prompt.
- Input utente: il testo del comando immesso nella riga di comando viene formattato come **user input**
- Testo sostituibile: il testo variabile, inclusi i nomi delle risorse scelte dall'utente o IDs generato da AWS servizi che è necessario includere nei comandi, è formattato come *replaceable text*. Nei comandi a più righe o nei comandi in cui è richiesto un input specifico da tastiera, i comandi da tastiera possono essere visualizzati anche come testo sostituibile.
- Output: l'output restituito dai AWS servizi viene visualizzato sotto l'input dell'utente e viene formattato come `computer output`

Il seguente esempio di **aws configure** comando mostra l'input dell'utente, il testo sostituibile e l'output:

1. Immettete **aws configure** nella riga di comando, quindi premete Invio.
2. Visualizza righe di testo in AWS CLI uscita, che richiedono l'immissione di informazioni aggiuntive.
3. Inserisci tutte le chiavi di accesso consecutivamente e quindi premi Invio.
4. Quindi, immettete il nome di una AWS regione nel formato mostrato, premete Invio, quindi premete Invio un'ultima volta per saltare l'impostazione del formato di output.
5. Il comando Invio finale viene mostrato come testo sostituibile in quanto per quella riga non esiste alcun input utente.

```
$ 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
```

L'esempio seguente mostra un semplice comando con l'output. Per usare questo esempio, immetti il testo completo del comando (il testo evidenziato dopo la richiesta) e quindi premi Invio. Il nome del gruppo di sicurezza, *mia-sg*, è sostituibile con il nome del gruppo di sicurezza desiderato. Il JSON documento, comprese le parentesi graffe arricciate, viene emesso. Se configuri l'output CLI in formato testo o tabella, l'output verrà formattato in modo diverso. [JSON](#) è il formato di output predefinito.

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

## Documentazione e risorse aggiuntive per AWS CLI

### AWS CLI documentazione e risorse

Oltre a questa guida per l'utente, le seguenti sono preziose risorse online per l'utilizzo di AWS CLI.

- [AWS CLI guida di riferimento alla versione 1](#)
- AWS CLI Archivio di [esempi di codice di scripting Bash](#). Esempi di script bash open source. Gli esempi di script in Bash sono ospitati nel [AWS Code Examples Repository](#) su GitHub
- [AWS CLI GitHub repository](#). È possibile visualizzare e eseguire il fork del codice sorgente per tutto il AWS CLI tempo. GitHub Unisciti alla community di utenti GitHubper fornire feedback, richiedere funzionalità e inviare i tuoi contributi. Ciò include la visualizzazione e la fornitura di esempi di comandi per la AWS CLI documentazione.
- [AWS CLI repository di esempi di alias](#) È possibile visualizzare ed eseguire il fork di esempi di AWS CLI alias su GitHub
- [AWS CLI versione 1 Changelog](#)
- [AWS CLI versione 2 Changelog](#)

### Altro e strumenti AWS SDKs

A seconda del caso d'uso, potresti voler scegliere uno dei AWS SDKs nostri strumenti più adatti alle tue esigenze:



- [AWS SDKse guida di riferimento agli strumenti](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [SDK AWS 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](#)
- [SDK AWS per SAP ABAP](#)
- [SDK AWS per Swift](#)
- [AWS Amplify](#)

# Installazione, aggiornamento e disinstallazione di AWS CLI

In questo argomento vengono forniti i collegamenti per installare, aggiornare e disinstallare la versione originale di AWS Command Line Interface (AWS CLI). La AWS CLI versione 1 è attualmente supportata, ma le nuove funzionalità aggiunte alla AWS CLI versione 2 potrebbero non essere aggiunte alla AWS CLI versione 1. Per utilizzare queste funzionalità, è necessario installare la AWS CLI versione 2. Per informazioni su come installare la versione 2, vedere [Installazione della AWS CLI versione 2](#).

AWS CLI istruzioni per l'installazione, l'aggiornamento e la disinstallazione:

- [Requisiti della versione di Python](#)
- [Installazione, aggiornamento e disinstallazione della AWS CLI versione 1 su Amazon Linux](#)
- [Installazione, aggiornamento e disinstallazione della AWS CLI versione 1 su Linux](#)
- [Installazione, aggiornamento e disinstallazione della AWS CLI versione 1 su macOS](#)
- [Installazione, aggiornamento e disinstallazione della AWS CLI versione 1 su Windows](#)
- [Installazione e aggiornamento della AWS CLI versione 1 in un ambiente virtuale](#)

## Requisiti della versione di Python

La AWS CLI versione 1 è stata creata utilizzando SDK for Python e pertanto richiede l'installazione di una versione compatibile di Python.

Matrice di supporto per la versione Python

AWS CLI versione	Versione Python supportata
1.32.0 — attuale	Python 3.8+
1.27.0 — 1,31.x	Python 3.7+
1.20.0 — 1,26.x	Python 3.6+
1.19.0 — 1.19x	Python 2.7+, Python 3.6+
1.17 — 1.18x	Python 2.7+, Python 3.4+

AWS CLI versione	Versione Python supportata
1,0 — 1,16x	Python 2.6 e versioni precedenti, Python 3.3 e precedenti

Per informazioni sulle ultime versioni di AWS CLI, consulta il [Changelog della AWS CLI versione 2](#) su. GitHub

## Installazione, aggiornamento e disinstallazione della AWS CLI versione 1 su Amazon Linux

La AWS CLI versione 1 è preinstallata su Amazon Linux e Amazon Linux 2. Controlla la versione attualmente installata utilizzando il comando seguente.

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

A seconda di quando hai creato l'istanza Amazon Linux, la AWS CLI versione 1 è preinstallata utilizzando uno dei seguenti gestori di pacchetti:

- [pip](#)
- [gnam](#)

## Prerequisiti

È necessario che Python 3.8 o versione successiva sia installato. Per istruzioni sull'installazione, consulta la pagina del [download di Python](#) nel manuale Beginner Guidedi Python.

Matrice di supporto per la versione Python

AWS CLI versione	Versione Python supportata
1.32.0 — attuale	Python 3.8+
1.27.0 — 1,31.x	Python 3.7+
1.20.0 — 1,26.x	Python 3.6+

AWS CLI versione	Versione Python supportata
1.19.0 — 1.19x	Python 2.7+, Python 3.6+
1.17 — 1.18x	Python 2.7+, Python 3.4+
1,0 — 1,16x	Python 2.6 e versioni precedenti, Python 3.3 e precedenti

## Installa, aggiorna o disinstalla usando pip

La maggior parte delle istanze Amazon Linux utilizza pip per preinstallare la AWS CLI versione 1.

### Installa o aggiorna la AWS CLI versione 1 su Amazon Linux usando pip

Per installare l'ultima versione della AWS CLI versione 1 per l'utente corrente, usa le seguenti istruzioni.

1. Se si dispone di Python versione 3 o successiva installato, consigliamo di utilizzare `pip3`. Utilizzare `pip3 install` per installare o aggiornare alla versione più recente della AWS CLI versione 1. Se si esegue il comando all'interno dell'[ambiente virtuale Python \(venv\)](#), non è necessario utilizzare l'opzione `--user`.

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

2. Verifica che la cartella che contiene `aws` faccia parte della variabile `PATH`.
  - a. Individua lo script del profilo della shell nella cartella utente. Se non hai la certezza di quale sia la tua shell, esegui `echo $SHELL`.

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

- Bash — `.bash_profile.profile`, o `.bash_login`
  - Zsh — `.zshrc`
  - Tcsh —, o `.tcshrc` `.cshrc` `.login`
- b. Aggiungi un comando di esportazione al termine dello script del profilo simile a quello dell'esempio seguente.

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

Questo comando inserisce il percorso, `$HOME/.local/bin` in questo esempio, all'inizio della variabile `$PATH` corrente.

c. Ricarica il profilo nella sessione corrente per rendere effettive tali modifiche.

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

3. Per verificare che sia in esecuzione la nuova versione, utilizzare il comando `aws --version`.

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

## Disinstalla la AWS CLI versione 1 usando pip

Se è necessario disinstallare il AWS CLI, utilizzare `pip uninstall`.

```
$ pip3 uninstall awscli
```

## Installa, aggiorna o disinstalla usando yum

La maggior parte delle istanze di Amazon Linux 2 utilizza yum per preinstallare la AWS CLI versione 1.

### Installa o aggiorna la AWS CLI versione 1 su Amazon Linux usando yum

Per installare l'ultima versione della AWS CLI versione 1 disponibile su Amazon Linux, esegui il seguente comando.

```
$ sudo yum install awscli
```

Per eseguire l'aggiornamento all'ultima versione della AWS CLI versione 1 disponibile su Amazon Linux, esegui il comando seguente.

```
$ sudo yum update awscli
```

Per verificare che tu stia utilizzando la versione più recente, usa il `aws --version` comando.

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

## Disinstalla la AWS CLI versione 1 usando yum

Per disinstallare il AWS CLI, usayum remove.

```
$ sudo yum remove awscli
```

## Risoluzione degli errori di AWS CLI installazione e disinstallazione

Se riscontri problemi dopo l'installazione o la disinstallazione di AWS CLI, consulta la procedura [Risolvere gli errori](#) per la risoluzione dei problemi. Per le procedure di risoluzione dei problemi più importanti, consulta [the section called "Errori relativi al comando non trovato"](#)[the section called "Il comando aws --version "" restituisce una versione diversa da quella installata"](#), e [the section called "Il comando "aws --version" restituisce una versione dopo la disinstallazione di AWS CLI"](#).

## Installazione, aggiornamento e disinstallazione della AWS CLI versione 1 su Linux

È possibile installare la AWS Command Line Interface (AWS CLI) versione 1 e le relative dipendenze sulla maggior parte delle distribuzioni Linux utilizzando il gestore di pacchetti o il pip programma di installazione in bundle.

Sebbene il awscli pacchetto sia disponibile negli archivi di altri gestori di pacchetti come apt andyum, questi non sono prodotti, gestiti o supportati da. AWS Si consiglia di installarlo solo AWS CLI dai punti di AWS distribuzione ufficiali, come documentato in questa guida.

### Sections

- [Prerequisiti](#)
- [Installa e disinstalla la AWS CLI versione 1 su Linux utilizzando il programma di installazione in bundle](#)
- [Installa e disinstalla la AWS CLI versione 1 usando pip](#)
- [Installa e disinstalla la AWS CLI versione 1 utilizzando Snapcraft](#)
- [Aggiungi l'eseguibile della AWS CLI versione 1 al percorso della riga di comando](#)
- [Risoluzione degli errori di AWS CLI installazione e disinstallazione](#)

## Prerequisiti

È necessario che Python 3.8 o versione successiva sia installato. Per istruzioni sull'installazione, consulta la pagina del [download di Python](#) nel manuale Beginner Guidedi Python.

Matrice di supporto per la versione Python

AWS CLI versione	Versione Python supportata
1.32.0 — attuale	Python 3.8+
1.27.0 — 1,31.x	Python 3.7+
1.20.0 — 1,26.x	Python 3.6+
1.19.0 — 1.19x	Python 2.7+, Python 3.6+
1.17 — 1.18x	Python 2.7+, Python 3.4+
1,0 — 1,16x	Python 2.6 e versioni precedenti, Python 3.3 e precedenti

## Installa e disinstalla la AWS CLI versione 1 su Linux utilizzando il programma di installazione in bundle

Su Linux o macOS, puoi utilizzare il programma di installazione in bundle per installare la versione 1 di AWS CLI. Il programma di installazione in bundle include tutte le dipendenze e può essere utilizzato offline.

### Note

Non supporta l'installazione in percorsi che contengono spazi.

### Argomenti

- [Installa la AWS CLI versione 1 utilizzando il programma di installazione in bundle con sudo](#)
- [Installa la AWS CLI versione 1 utilizzando il programma di installazione fornito senza sudo](#)
- [Disinstalla il programma di installazione in bundle della AWS CLI versione 1](#)

## Installa la AWS CLI versione 1 utilizzando il programma di installazione in bundle con **sudo**

I passaggi seguenti consentono di installare la AWS CLI versione 1 dalla riga di comando su qualsiasi build di Linux o macOS.

Qui di seguito è riportato un riepilogo dei comandi di installazione con le spiegazioni che puoi tagliare e incollare per eseguirli come un singolo set di comandi.

Per la versione più recente di AWS CLI, usa il seguente blocco di comandi:

```
$ 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
```

Per una versione specifica di AWS CLI, aggiungi un trattino e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione **1.16.312** `awscli-bundle-1.16.312.zip` risulterebbe nel seguente comando:

```
$ 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
```

Segui questi passaggi dalla riga di comando per installare la AWS CLI versione 1 utilizzando il programma di installazione in bundle.

Per installare la AWS CLI versione 1 utilizzando il programma di installazione in dotazione

1. Scarica il programma di installazione in bundle della AWS CLI versione 1 utilizzando uno dei seguenti metodi.

- Download con il comando `curl`.

Per la versione più recente di AWS CLI, utilizzate il seguente blocco di comandi:

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



Per una versione specifica di AWS CLI, aggiungi un trattino e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione **1.16.312** `awscli-bundle-1.16.312.zip` risulterebbe nel seguente comando:

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

- Scarica utilizzando il collegamento diretto.

Per la versione più recente di AWS CLI: <https://s3.amazonaws.com/aws-cli/awscli-bundle.zip>

Per una versione specifica di AWS CLI, aggiungi un trattino e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione **1.16.312** `awscli-bundle-1.16.312.zip` risulterebbe nel seguente URL <https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip>

2. Estrarre i file dal pacchetto. Se non disponi di `unzip` per l'estrazione dei file, usa il programma di gestione dei pacchetti integrato della distribuzione Linux per eseguire l'installazione.

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

3. Esegui il programma di installazione. L'installatore installa AWS CLI at `/usr/local/aws` e crea il collegamento simbolico `aws` nella directory `/usr/local/bin`. Utilizzando l'opzione `-b` per creare un link simbolico, si elimina la necessità di specificare la directory di installazione nella variabile `$PATH` dell'utente. Ciò dovrebbe consentire a tutti gli utenti di richiamarli `aws` entrando AWS CLI da qualsiasi directory.

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

Per default, lo script di installazione viene eseguito con la versione di Python di default del sistema. Se hai installato una versione alternativa di Python e desideri utilizzare quella versione per installarla AWS CLI, esegui lo script di installazione con quella versione tramite il percorso assoluto dell'eseguibile Python, come segue.

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

4. Verifica che sia AWS CLI installata correttamente.

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

Se si verifica un errore, consulta [Risoluzione degli errori per AWS CLI](#).

## Installa la AWS CLI versione 1 utilizzando il programma di installazione fornito senza **sudo**

Se non disponi sudo delle autorizzazioni o desideri installarlo AWS CLI solo per l'utente corrente, puoi utilizzare una versione modificata dei comandi precedenti. I primi due comandi sono gli stessi.

Per la versione più recente di AWS CLI, usa il seguente blocco di comandi:

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

Per una versione specifica di AWS CLI, aggiungi un trattino e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione **1.16.312** `awscli-bundle-1.16.312.zip` risulterebbe nel seguente comando:

```
$ 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
```

Per installare la AWS CLI versione 1 per l'utente corrente

1. Scarica il programma di installazione in bundle della AWS CLI versione 1 in uno dei seguenti modi.
  - Download con il comando `curl`.

Per la versione più recente di AWS CLI, utilizzate il seguente blocco di comandi:

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

Per una versione specifica di AWS CLI, aggiungi un trattino e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione **1.16.312** `awscli-bundle-1.16.312.zip` risulterebbe nel seguente comando:

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

- Scarica utilizzando il collegamento diretto.

Per la versione più recente di AWS CLI: <https://s3.amazonaws.com/aws-cli/awscli-bundle.zip>

Per una versione specifica di AWS CLI, aggiungi un trattino e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione **1.16.312** `awscli-bundle-1.16.312.zip` risulterebbe nel seguente URL <https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip>

2. Estrarre i file dal pacchetto utilizzando `unzip`. Se non disponi di `unzip`, usa il programma di gestione dei pacchetti integrato della distribuzione Linux per eseguire l'installazione.

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

3. Esegui il programma di installazione. L'installatore installa AWS CLI at `/usr/local/aws` e crea il collegamento simbolico `aws` nella directory. `/usr/local/bin` L'ultimo comando utilizza il parametro `-b` per specificare la cartella in cui il programma di installazione posiziona il file di collegamento simbolico `aws`. È necessario disporre delle autorizzazioni di scrittura nella cartella specificata.

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

Questo AWS CLI installa il nella posizione predefinita (`~/local/lib/aws`) e crea un collegamento simbolico (link simbolico) in. `~/bin/aws` Per il corretto funzionamento del link simbolico, `~/bin` deve trovarsi nella variabile di ambiente `PATH`.

```
$ 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. Assicurati che la cartella contenente la AWS CLI versione 1 faccia parte della tua variabile. `PATH`

- a. Trova lo script del profilo della tua shell nella tua cartella utente. Se non hai la certezza di quale sia la tua shell, esegui `echo $SHELL`.

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

- Bash — `.bash_profile.profile`, o `.bash_login`
- Zsh — `.zshrc`
- Tcsh —, o `.tcshrc .cshrc .login`

- b. Aggiungi un comando di esportazione al termine dello script del profilo simile a quello dell'esempio seguente.

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

Questo comando inserisce il percorso, `~/.local/bin` in questo esempio, all'inizio della variabile `PATH` corrente.

- c. Ricarica il profilo nella sessione corrente per rendere effettive tali modifiche.

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

5. Verifica che sia AWS CLI installato correttamente.

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

Se si verifica un errore, consulta [Risoluzione degli errori per AWS CLI](#).

## Disinstalla il programma di installazione in bundle della AWS CLI versione 1

1. Se lo hai installato AWS CLI utilizzando il programma di installazione in bundle, segui queste istruzioni. Il programma di installazione in bundle non inserisce nulla al di fuori della directory di installazione, a eccezione del link simbolico opzionale, pertanto per effettuare la disinstallazione è sufficiente eliminare questi due elementi.

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

2. (Facoltativo) Rimuovi le informazioni condivise AWS SDK e AWS CLI le impostazioni nella `.aws` cartella.

#### Warning

Queste impostazioni di configurazione e credenziali sono condivise tra tutti AWS SDKs e tra. AWS CLI Se rimuovi questa cartella, non sarà più possibile accedervi da nessuno AWS SDKs che sia ancora presente nel sistema.

La posizione predefinita della `.aws` cartella varia a seconda delle piattaforme, per impostazione predefinita la cartella si trova in `~/.aws/`. Se l'utente dispone dell'autorizzazione di scrittura per questa directory, non è necessario utilizzarla `sudo`.

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

## Installa e disinstalla la AWS CLI versione 1 usando pip

### Argomenti

- [Installazione pip.](#)
- [Installa e aggiorna la AWS CLI versione 1 usando pip](#)
- [Disinstalla AWS CLI usando pip](#)

### Installazione pip.

Se non disponi già di `pip` installato, puoi installarlo utilizzando lo script fornito da Python Packaging Authority. Esegui `pip --version` per vedere se la versione di Linux di cui disponi include già Python e `pip`. Se disponi di Python versione 3 o successiva installato, ti consigliamo di utilizzare il comando `pip3`.

1. Utilizza il comando `curl` per scaricare lo script di installazione. Il comando seguente utilizza il parametro `-O` (lettera maiuscola O) per specificare che il file scaricato deve essere archiviato nella directory corrente utilizzando lo stesso nome che ha sull'host remoto.

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

2. Esegui lo script con il comando `python` o `python3` per scaricare e installare la versione più recente di `pip` e gli altri pacchetti di supporto necessari. Quando includi l'opzione `--user`, lo script installa `pip` nel percorso `~/local/bin`.

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

3. Assicurati che la directory che contiene faccia `pip` parte della tua `PATH` variabile.
  - a. Trova lo script del profilo della tua shell nella tua cartella utente. Se non hai la certezza di quale sia la tua shell, esegui `echo $SHELL`.

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

- Bash — `.bash_profile`, `.profile`, o `.bash_login`
  - Zsh — `.zshrc`
  - Tcsh —, o `.tcshrc`, `.cshrc`, `.login`
- b. Aggiungi un comando di esportazione al termine dello script del profilo simile a quello dell'esempio seguente.

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

Questo comando inserisce il percorso, `~/local/bin` in questo esempio, all'inizio della variabile `PATH` corrente.

- c. Ricarica il profilo nella sessione corrente per rendere effettive tali modifiche.

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

4. Per verificare che `pip` o `pip3` sia installato correttamente, eseguire il comando seguente.

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

## Installa e aggiorna la AWS CLI versione 1 usando pip

1. Utilizzare il comando `pip` o `pip3` per installare o aggiornare l' AWS CLI. Se disponi di Python versione 3 o successiva, ti consigliamo di utilizzare il comando `pip3`. Lo `--user` switch, `pip` installa il AWS CLI in `~/.local/bin`

Per la versione più recente di AWS CLI, usa il seguente blocco di comandi:

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

Per una versione specifica di AWS CLI, aggiungi due segni di uguale `=` e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione `1.16.312` sarebbe `==1.16.312` risultante nel seguente comando:

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

### Note

Utilizzate le regole di quotazione appropriate per il vostro terminale. Per utilizzare il `=` carattere, potrebbe essere necessario utilizzare virgolette singole o doppie per scappare correttamente. L'esempio seguente evade utilizzando virgolette singole:

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

2. Verificare che sia AWS CLI installato correttamente.

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

Se si verifica un errore, consulta [Risoluzione degli errori per AWS CLI](#).

## Disinstalla AWS CLI usando pip

1. Se hai installato la AWS CLI versione 1 utilizzando `pip`, devi anche disinstallarla utilizzando `pip`.

```
$ pip uninstall awscli
```

Se si utilizza la versione Python 2 o 3, potrebbe essere necessario utilizzare il comando `pip2` o `pip3`. Usa il `aws --version` comando per determinare la versione di Python associata alla versione 1 installata AWS CLI .

```
$ pip3 uninstall awscli
```

Potrebbe essere necessario riavviare la finestra del prompt dei comandi o il computer per rimuovere tutti i file.

2. (Facoltativo) Rimuovi le informazioni condivise AWS SDK e AWS CLI le impostazioni nella `.aws` cartella.

#### Warning

Queste impostazioni di configurazione e credenziali sono condivise tra tutti AWS SDKs e tra. AWS CLI Se rimuovi questa cartella, non sarà più possibile accedervi da nessuno AWS SDKs che sia ancora presente nel sistema.

La posizione predefinita della `.aws` cartella varia a seconda delle piattaforme, per impostazione predefinita la cartella si trova in `~/.aws/`. Se l'utente dispone dell'autorizzazione di scrittura per questa directory, non è necessario utilizzarla `sudo`.

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

## Installa e disinstalla la AWS CLI versione 1 utilizzando Snapcraft

### Argomenti

- [Installa snap](#)
- [Installa e aggiorna la AWS CLI versione 1 usando snap](#)
- [Disinstalla AWS CLI usando snap](#)

### Installa snap

Se non l'hai già `snap` installato, puoi installarlo utilizzando le istruzioni fornite da Canonical Snapcraft. Esegui `snap version` per vedere se la tua versione di Linux lo include già. `snap`



1. Installa Snapcraft sulla tua piattaforma. Per informazioni sull'installazione di Snapcraft, consulta [Installazione del demone](#) nella documentazione di Snap.
2. Riavvia il sistema in modo che le PATH variabili vengano aggiornate correttamente. Se riscontri problemi di installazione, segui i passaggi in [Risolvere i problemi comuni](#) nella documentazione di Snap.
3. Per verificare che snap sia installato correttamente, esegui il comando seguente.

```
$ snap version
```

## Installa e aggiorna la AWS CLI versione 1 usando snap

1. Esegui il seguente `snap install` comando per la AWS CLI versione 1.

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

A seconda delle autorizzazioni di cui disponi, potrebbe essere necessario aggiungere `sudo` qualcosa al comando.

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

2. Verifica che sia AWS CLI installato correttamente.

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

Se si verifica un errore, consulta [Risoluzione degli errori per AWS CLI](#).

## Disinstalla AWS CLI usando snap

1. Se hai installato la AWS CLI versione 1 utilizzando `snap`, devi anche disinstallarla utilizzando `snap`.

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

Potrebbe essere necessario riavviare la finestra del prompt dei comandi o il computer per rimuovere tutti i file.

2. (Facoltativo) Rimuovi le informazioni condivise AWS SDK e AWS CLI le impostazioni nella `.aws` cartella.

**⚠ Warning**

Queste impostazioni di configurazione e credenziali sono condivise tra tutti AWS SDKs e tra. AWS CLI Se rimuovi questa cartella, non sarà più possibile accedervi da nessuno AWS SDKs che sia ancora presente nel sistema.

La posizione predefinita della `.aws` cartella varia a seconda delle piattaforme, per impostazione predefinita la cartella si trova in `~/.aws/`. Se disponi dei permessi di scrittura per questa directory, non è necessario sudo utilizzarla.

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

## Aggiungi l'eseguibile della AWS CLI versione 1 al percorso della riga di comando

Dopo l'installazione con `pip orsnap`, potrebbe essere necessario aggiungere l'awseseguibile alla variabile di PATH ambiente del sistema operativo.

È possibile verificare AWS CLI in quale cartella è pip installato il file eseguendo il comando seguente.

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

Per fare riferimento, puoi utilizzare `~/.local/bin/` perché `/home/username` corrisponde a `~` in Linux.

Se hai omesso l'opzione `--user` e pertanto non hai eseguito l'installazione in modalità utente, l'eseguibile potrebbe essere nella cartella `bin` dell'installazione Python. Se non conosci il percorso di installazione di Python, esegui questo comando.

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

L'output potrebbe essere il percorso di un link simbolico e non dell'eseguibile vero e proprio. Esegui `ls -al` per vedere a cosa punta.

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

pip installa i programmi nella stessa cartella che contiene l'applicazione Python. Aggiungi questa cartella alla variabile PATH.

Per modificare la variabile **PATH**

1. Individua lo script del profilo della shell nella cartella utente. Se non hai la certezza di quale sia la tua shell, esegui `echo $SHELL`.

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

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

2. Aggiungi un comando di esportazione allo script del tuo profilo.

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

Questo comando aggiunge un percorso, in questo esempio `~/.local/bin`, per la variabile PATH corrente.

3. Carica il profilo aggiornato nella sessione corrente.

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

## Risoluzione degli errori di AWS CLI installazione e disinstallazione

Se riscontri problemi dopo l'installazione o la disinstallazione di AWS CLI, consulta la procedura [Risolvere gli errori](#) per la risoluzione dei problemi. Per le procedure di risoluzione dei problemi più importanti, consulta [the section called “Errori relativi al comando non trovato”](#) [the section called “Il](#)

[comando `aws --version` restituisce una versione diversa da quella installata](#), e [the section called "Il comando `aws --version` restituisce una versione dopo la disinstallazione di AWS CLI"](#).

## Installazione, aggiornamento e disinstallazione della AWS CLI versione 1 su macOS

Puoi installare la AWS Command Line Interface (AWS CLI) versione 1 e le sue dipendenze su macOS utilizzando il programma di installazione in bundle o. pip

### Sections

- [Prerequisiti](#)
- [Installa, aggiorna e disinstalla la AWS CLI versione 1 su macOS utilizzando il programma di installazione in bundle](#)
- [Installa, aggiorna e disinstalla la AWS CLI versione 1 usando pip](#)
- [Risoluzione degli errori di AWS CLI installazione e disinstallazione](#)

## Prerequisiti

Prima di poter installare la AWS CLI versione 1 su macOS, assicurati di avere installato Python 3.8 o versione successiva. Per istruzioni sull'installazione, consulta la pagina del [download di Python](#) nel manuale Beginner Guidedi Python.

Matrice di supporto per la versione Python

AWS CLI versione	Versione Python supportata
1.32.0 — attuale	Python 3.8+
1.27.0 — 1,31.x	Python 3.7+
1.20.0 — 1,26.x	Python 3.6+
1.19.0 — 1.19x	Python 2.7+, Python 3.6+
1.17 — 1.18x	Python 2.7+, Python 3.4+

AWS CLI versione	Versione Python supportata
1,0 — 1,16x	Python 2.6 e versioni precedenti, Python 3.3 e precedenti

## Installa, aggiorna e disinstalla la AWS CLI versione 1 su macOS utilizzando il programma di installazione in bundle

Su Linux o macOS, puoi usare il programma di installazione in bundle per installare la versione 1 di (). AWS Command Line Interface AWS CLI Il programma di installazione in bundle include tutte le dipendenze e può essere utilizzato offline.

Non supporta l'installazione in percorsi che contengono spazi.

### Argomenti

- [Installa la AWS CLI versione 1 utilizzando il programma di installazione in dotazione con sudo](#)
- [Installa la AWS CLI versione 1 utilizzando il programma di installazione in bundle senza sudo](#)
- [Disinstalla il programma di installazione in bundle della AWS CLI versione 1](#)

## Installa la AWS CLI versione 1 utilizzando il programma di installazione in dotazione con **sudo**

I passaggi seguenti consentono di installare la AWS CLI versione 1 dalla riga di comando su qualsiasi build di macOS.

Il seguente è un riepilogo dei comandi di installazione che puoi tagliare e incollare per eseguirli come un singolo set di comandi.

Per la versione più recente di AWS CLI, usa il seguente blocco di comandi:

```
$ 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
```

Per una versione specifica di AWS CLI, aggiungi un trattino e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione **1.16.312** `awscli-bundle-1.16.312.zip` risulterebbe nel seguente comando:

```
$ 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
```

Per installare la AWS CLI versione 1 utilizzando il programma di installazione in dotazione

1. Scarica il programma di installazione in bundle della AWS CLI versione 1 in uno dei seguenti modi:
  - Download con il comando `curl`.

Per la versione più recente di AWS CLI, utilizzate il seguente blocco di comandi:

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

Per una versione specifica di AWS CLI, aggiungi un trattino e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione `1.16.312` `awscli-bundle-1.16.312.zip` risulterebbe nel seguente comando:

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

- Scarica utilizzando il collegamento diretto.

Per la versione più recente di AWS CLI: <https://s3.amazonaws.com/aws-cli/awscli-bundle.zip>

Per una versione specifica di AWS CLI, aggiungi un trattino e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione `1.16.312` `awscli-bundle-1.16.312.zip` risulterebbe nel seguente URL <https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip>

2. Estrazione (decompressione) dei file dal pacchetto. Se non lo hai `unzip`, usa il gestore di pacchetti integrato nella tua macOS distribuzione per installarlo.

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

3. Esegui il programma di installazione. L'installatore installa AWS CLI at `/usr/local/aws` e crea il collegamento simbolico `aws` nella cartella `/usr/local/bin` Utilizzando l'opzione `-b`

per creare un link simbolico, si elimina la necessità di specificare la cartella di installazione nella variabile \$PATH dell'utente. Ciò dovrebbe consentire a tutti gli utenti di richiamarli aws entrando AWS CLI da qualsiasi directory.

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

Per default, lo script di installazione viene eseguito con la versione di Python di default del sistema. Se hai installato una versione alternativa di Python e desideri utilizzarla per installarla AWS CLI, esegui lo script di installazione con quella versione tramite il percorso assoluto dell'eseguibile Python, come segue.

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

4. Verifica che sia AWS CLI installato correttamente.

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

Se si verifica un errore, consulta [Risoluzione degli errori per AWS CLI](#).

## Installa la AWS CLI versione 1 utilizzando il programma di installazione in bundle senza **sudo**

Se non disponi sudo delle autorizzazioni o desideri installarlo AWS CLI solo per l'utente corrente, puoi utilizzare una versione modificata dei comandi precedenti. I primi due comandi sono gli stessi.

Per la versione più recente di AWS CLI, usa il seguente blocco di comandi:

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

Per una versione specifica di AWS CLI, aggiungi un trattino e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione **1.16.312** awscli-bundle-1.16.312.zip risulterebbe nel seguente comando:

```
$ 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
```

Per installare la AWS CLI versione 1 per l'utente corrente

1. Scarica il programma di installazione in bundle della AWS CLI versione 1 utilizzando uno dei seguenti metodi:

- Download con il comando `curl`.

Per la versione più recente di AWS CLI, utilizzate il seguente blocco di comandi:

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

Per una versione specifica di AWS CLI, aggiungi un trattino e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione **1.16.312** `awscli-bundle-1.16.312.zip` risulterebbe nel seguente comando:

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

- Scarica utilizzando il collegamento diretto.

Per la versione più recente di AWS CLI: <https://s3.amazonaws.com/aws-cli/awscli-bundle.zip>

Per una versione specifica di AWS CLI, aggiungi un trattino e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione **1.16.312** `awscli-bundle-1.16.312.zip` risulterebbe nel seguente URL <https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip>

2. Estrarre i file dal pacchetto. Se non disponi di `unzip`, usa il programma di gestione dei pacchetti integrato della distribuzione Linux per eseguire l'installazione.

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

3. Esegui il programma di installazione. L'installatore installa AWS CLI at `/usr/local/aws` e crea il collegamento simbolico `aws` nella directory `/usr/local/bin`. L'ultimo comando utilizza il parametro `-b` per specificare la cartella in cui il programma di installazione posiziona il file di



collegamento simbolico `aws`. È necessario disporre delle autorizzazioni di scrittura nella directory specificata.

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

Questo AWS CLI installa il nella posizione predefinita (`~/local/lib/aws`) e crea un collegamento simbolico (link simbolico) in `~/bin/aws`. Per il corretto funzionamento del link simbolico, `~/bin` deve trovarsi nella variabile di ambiente `$PATH`.

```
$ 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. Assicurati che la cartella in cui è installata la AWS CLI versione 1 faccia parte della tua variabile `$PATH`

a. Trova lo script del profilo della tua shell nella tua cartella utente. Se non hai la certezza di quale sia la tua shell, esegui `echo $SHELL`.

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

- Bash — `.bash_profile`, `.profile`, o `.bash_login`
- Zsh — `.zshrc`
- Tcsh —, o `.tcshrc`, `.cshrc`, `.login`

b. Aggiungi un comando di esportazione al termine dello script del profilo simile a quello dell'esempio seguente.

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

Questo comando inserisce il percorso, `~/local/bin` in questo esempio, all'inizio della variabile `PATH` corrente.

c. Ricarica il profilo nella sessione corrente per rendere effettive tali modifiche.

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

5. Verifica che sia AWS CLI stato installato correttamente.

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

Se si verifica un errore, consulta [Risoluzione degli errori per AWS CLI](#).

## Disinstalla il programma di installazione in bundle della AWS CLI versione 1

1. Il programma di installazione in bundle inserisce tutto all'interno della directory di installazione tranne il collegamento simbolico opzionale, quindi per disinstallare, è sufficiente eliminare quei due elementi.

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

2. (Facoltativo) Rimuovi le informazioni condivise AWS SDK e AWS CLI le impostazioni nella `.aws` cartella.

### Warning

Queste impostazioni di configurazione e credenziali sono condivise tra tutti AWS SDKs e tra. AWS CLI Se rimuovi questa cartella, non sarà più possibile accedervi da nessuno AWS SDKs che sia ancora presente nel sistema.

La posizione predefinita della `.aws` cartella varia a seconda delle piattaforme, per impostazione predefinita la cartella si trova in `~/.aws/`. Se l'utente dispone dell'autorizzazione di scrittura per questa directory, non è necessario utilizzarla `sudo`.

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

## Installa, aggiorna e disinstalla la AWS CLI versione 1 usando pip

È possibile utilizzare `pip` direttamente per installare AWS CLI.

### Argomenti

- [Installazione pip](#).

- [Installa e aggiorna AWS CLI usando pip](#)
- [Aggiungi l'eseguibile della AWS CLI versione 1 al percorso della riga di comando di macOS](#)
- [Disinstalla usando pip AWS CLI](#)

## Installazione pip.

Se non disponi già di pip installato, puoi installarlo utilizzando lo script fornito da Python Packaging Authority. Esegui `pip --version` per vedere se la versione di Linux di cui disponi include già Python e pip. Se disponi di Python versione 3 o successiva installato, ti consigliamo di utilizzare il comando `pip3`.

1. Utilizza il comando `curl` per scaricare lo script di installazione. Il comando seguente utilizza il parametro `-O` (lettera maiuscola O) per specificare che il file scaricato deve essere archiviato nella cartella corrente utilizzando lo stesso nome che ha sull'host remoto.

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

2. Esegui lo script con il comando `python` o `python3` per scaricare e installare la versione più recente di pip e gli altri pacchetti di supporto necessari. Quando includi l'opzione `--user`, lo script installa pip nel percorso `~/.local/bin`.

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

## Installa e aggiorna AWS CLI usando pip

1. Utilizzare il comando `pip` o `pip3` per installare AWS CLI. Se disponi di Python versione 3 successiva, ti consigliamo di utilizzare il comando `pip3`.

Per la versione più recente di AWS CLI, usa il seguente blocco di comandi:

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

Per una versione specifica di AWS CLI, aggiungi due segni di uguale `=` e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione **1.16.312** sarebbe **`==1.16.312`** risultante nel seguente comando:

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

### Note

Utilizzate le regole di quotazione appropriate per il vostro terminale. Per utilizzare il = carattere, potrebbe essere necessario utilizzare virgolette singole o doppie per scappare correttamente. L'esempio seguente evade utilizzando virgolette singole:

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

2. Verificare che AWS CLI sia installato correttamente.

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

Se il programma non è presente, [aggiungilo al percorso della riga di comando](#).

## Aggiungi l'eseguibile della AWS CLI versione 1 al percorso della riga di comando di macOS

Dopo avere eseguito l'installazione con `pip`, potrebbe essere necessario aggiungere il programma `aws` alla variabile di ambiente `PATH` del sistema operativo in uso. La posizione del programma dipende dal percorso di installazione di Python.

Example AWS CLI posizione di installazione - macOS con Python 3.6 e (modalità utente) `pip`

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

Sostituisci la versione di Python di cui disponi con la versione dell'esempio precedente.

Se non conosci il percorso di installazione di Python, esegui `which python`.

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

L'output potrebbe essere il percorso a un link simbolico, non il programma effettivo. Esegui `ls -al` per vedere a cosa punta.

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

pip installa i programmi nella stessa cartella che contiene l'applicazione Python. Aggiungi questa cartella alla variabile PATH.

Per modificare la variabile **PATH**

1. Individua lo script del profilo della shell nella cartella utente. Se non hai la certezza di quale sia la tua shell, esegui echo \$SHELL.

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

- Bash – .bash\_profile, .profile, o .bash\_login
  - Zsh – .zshrc
  - Tcsh – .tcshrc, .cshrc, o .login
2. Aggiungi un comando di esportazione allo script del tuo profilo.

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

Questo comando aggiunge un percorso, in questo esempio ~/.local/bin, per la variabile PATH corrente.

3. Carica il profilo aggiornato nella sessione corrente.

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

## Disinstalla usando pip AWS CLI

1. Se hai installato la AWS CLI versione 1 utilizzando pip, devi anche disinstallarla utilizzando pip.

```
$ pip uninstall awscli
```

Se si utilizza la versione Python 2 o 3, potrebbe essere necessario utilizzare il comando pip2 o pip3. Usa il aws --version comando per determinare la versione di Python associata alla versione 1 installata AWS CLI .

```
$ pip3 uninstall awscli
```

Potrebbe essere necessario riavviare la finestra del prompt dei comandi o il computer per rimuovere tutti i file.

2. (Facoltativo) Rimuovi le informazioni condivise AWS SDK e AWS CLI le impostazioni nella `.aws` cartella.

#### Warning

Queste impostazioni di configurazione e credenziali sono condivise tra tutti AWS SDKs e tra. AWS CLI Se rimuovi questa cartella, non sarà più possibile accedervi da nessuno AWS SDKs che sia ancora presente nel sistema.

La posizione predefinita della `.aws` cartella varia a seconda delle piattaforme, per impostazione predefinita la cartella si trova in `~/.aws/`. Se l'utente dispone dell'autorizzazione di scrittura per questa directory, non è necessario utilizzarla `sudo`.

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

## Risoluzione degli errori di AWS CLI installazione e disinstallazione

Se riscontri problemi dopo l'installazione o la disinstallazione di AWS CLI, consulta la procedura [Risolvere gli errori](#) per la risoluzione dei problemi. Per le procedure di risoluzione dei problemi più importanti, consulta [the section called "Errori relativi al comando non trovato"](#) [the section called "Il comando `aws --version` restituisce una versione diversa da quella installata"](#), e [the section called "Il comando `aws --version` restituisce una versione dopo la disinstallazione di AWS CLI"](#).

## Installazione, aggiornamento e disinstallazione della AWS CLI versione 1 su Windows

Puoi installare la versione 1 di AWS Command Line Interface (AWS CLI) su Windows utilizzando un programma di installazione autonomo (consigliato) o `pip`, che è un gestore di pacchetti per Python.

Quando digiti un comando, non includere il simbolo del prompt (C:\>). Il simbolo è incluso negli elenchi di programma per differenziare i comandi digitati dall'utente dall'output restituito dall' AWS CLI. Nella parte rimanente di questa guida viene utilizzato il simbolo del prompt generico, \$, salvo nei casi in cui il comando sia specifico di Windows.

## Argomenti

- [Installa, aggiorna e disinstalla la AWS CLI versione 1 utilizzando il programma di installazione MSI](#)
- [Installa, aggiorna e disinstalla la AWS CLI versione 1 usando Python e pip su Windows](#)
- [Aggiungi l'eseguibile della AWS CLI versione 1 al percorso della riga di comando](#)
- [Risoluzione degli errori di AWS CLI installazione e disinstallazione](#)

## Installa, aggiorna e disinstalla la AWS CLI versione 1 utilizzando il programma di installazione MSI

La AWS CLI versione 1 è supportata in Windows XP o versioni successive. Per gli utenti Windows, il pacchetto di MSI installazione offre un modo semplice e intuitivo per installare la AWS CLI versione 1 senza installare altri prerequisiti.

### Installa e aggiorna la AWS CLI versione 1 utilizzando il MSI programma di installazione

Controlla la pagina [Releases](#) GitHub per vedere quando è stata rilasciata l'ultima versione. Quando vengono rilasciati gli aggiornamenti, è necessario ripetere il processo di installazione per ottenere l'ultima versione della AWS CLI versione 1.

1. Scarica il programma di MSI installazione appropriato:
  - AWS CLI MSI [programma di installazione per Windows \(64 bit\): .msi https://s3.amazonaws.com/aws-cli/AWSCLI64PY3](https://s3.amazonaws.com/aws-cli/AWSCLI64PY3)
  - AWS CLI MSI [programma di installazione per Windows \(32 bit\): .msi https://s3.amazonaws.com/aws-cli/AWSCLI32PY3](https://s3.amazonaws.com/aws-cli/AWSCLI32PY3)
  - AWS CLI file di installazione combinato per Windows: <https://s3.amazonaws.com/aws-cli/AWSCLISetup.exe> (include i programmi di installazione a 32 e 64 bit e MSI installa automaticamente la versione corretta)
2. Esegui il programma di MSI installazione scaricato o il file di installazione.

3. Seguire le istruzioni su schermo. Per impostazione predefinita, la AWS CLI versione 1 viene installata su C:\Program Files\Amazon\AWSCLI (versione a 64 bit) o C:\Program Files (x86)\Amazon\AWSCLI (versione a 32 bit).
4. Per confermare l'installazione, utilizza il comando `aws --version` al prompt dei comandi (apri il menu Start e cerca cmd per avviare il prompt dei comandi).

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

Se Windows non riesce a trovare il programma, potrebbe essere necessario chiudere e riaprire il prompt dei comandi per aggiornare il percorso o [aggiungere manualmente la directory di installazione alla variabile](#) di ambiente. PATH

## Disinstalla la versione 1 AWS CLI

Per utilizzare le seguenti istruzioni di disinstallazione, è necessario aver installato la AWS CLI versione 1 con il programma di MSI installazione o il file di configurazione.

1. Aprire Programmi e caratteristiche eseguendo una delle seguenti operazioni:
  - Aprire il Pannello di controllo e scegliere Programmi e caratteristiche.
  - Aprire un prompt dei comandi e inserire il comando seguente.

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

2. Seleziona la voce denominata AWS Command Line Interface e quindi scegli Uninstall (Disinstalla) per avviare il programma di disinstallazione.
3. Conferma di voler disinstallare il AWS CLI.
4. (Facoltativo) Rimuovi le informazioni condivise AWS SDK e AWS CLI le impostazioni nella `.aws` cartella.

### Warning

Queste impostazioni di configurazione e credenziali sono condivise tra tutti AWS SDKs e tra. AWS CLI Se rimuovi questa cartella, non sarà più possibile accedervi da nessuno AWS SDKs che sia ancora presente nel sistema.



La posizione predefinita della `.aws` cartella varia a seconda delle piattaforme, per impostazione predefinita la cartella si trova in `%UserProfile%\.aws`.

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

## Installa, aggiorna e disinstalla la AWS CLI versione 1 usando Python e pip su Windows

Python Software Foundation offre programmi di installazione per Windows che includono pip.

### Prerequisiti

È necessario che Python 3.8 o versione successiva sia installato. Per istruzioni sull'installazione, consulta la pagina del [download di Python](#) nel manuale Beginner Guidedi Python.

### Installa e aggiorna la AWS CLI versione 1 usando pip

1. Per installare la AWS CLI versione 1, usa il `pip3` comando (se usi Python versione 3 o successiva) o il `pip` comando.

Per la versione più recente di AWS CLI, usa il seguente blocco di comandi:

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

Per una versione specifica di AWS CLI, aggiungi un simbolo minore di `<` e il numero di versione al nome del file. Per questo esempio, il nome del file per la versione `1.16.312` sarebbe `<1.16.312` risultante nel seguente comando:

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

2. Verificare che la AWS CLI versione 1 sia installata correttamente. Se non c'è risposta, consulta la sezione [Aggiungi l'eseguibile della AWS CLI versione 1 al percorso della riga di comando](#).

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

## Disinstalla la AWS CLI versione 1 usando pip

1. Se hai installato la AWS CLI versione 1 utilizzando pip, devi anche disinstallarla utilizzando pip.

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

Se si utilizza la versione Python 2 o 3, potrebbe essere necessario utilizzare il comando `pip2` o `pip3`. Usa il `aws --version` comando per determinare la versione di Python associata alla versione 1 installata AWS CLI .

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

Potrebbe essere necessario riavviare la finestra del prompt dei comandi o il computer per rimuovere tutti i file.

2. (Facoltativo) Rimuovi le informazioni condivise AWS SDK e AWS CLI le impostazioni nella `.aws` cartella.

### Warning

Queste impostazioni di configurazione e credenziali sono condivise tra tutti AWS SDKs e tra. AWS CLI Se rimuovi questa cartella, non sarà più possibile accedervi da nessuno AWS SDKs che sia ancora presente nel sistema.

La posizione predefinita della `.aws` cartella varia a seconda delle piattaforme, per impostazione predefinita la cartella si trova in `%UserProfile%\aws`.

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

## Aggiungi l'eseguibile della AWS CLI versione 1 al percorso della riga di comando

Dopo aver installato la AWS CLI versione 1 con pip, aggiungi il aws programma alla variabile di PATH ambiente del tuo sistema operativo. Con un'MSI installazione, ciò dovrebbe avvenire automaticamente. Ma se il comando aws non viene eseguito dopo l'installazione, potrebbe essere necessario impostarlo manualmente.

1. Utilizzare il comando `where` per trovare la posizione del file `aws`. Per impostazione predefinita, il comando `where` mostra la posizione di un programma specificato nel PATH del sistema.

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

I percorsi che appaiono dipendono dalla piattaforma e dal metodo utilizzato per installare l' AWS CLI. I nomi delle cartelle che includono i numeri di versione possono variare. Questi esempi riflettono l'uso di Python versione 3.7. Sostituire la versione con il numero di versione che si sta utilizzando, in base alle necessità. I percorsi tipici includono quanto segue:

- Python 3 e `pip3` `C:\Program Files\Python37\Scripts\`
- Python 3 e **pip3** l'opzione `--user` nelle versioni precedenti di Windows — `%USERPROFILE%\AppData\Local\Programs\Python\Python37\Scripts`
- Python 3 e **pip3** l'opzione `--user` su Windows 10 — `%USERPROFILE%\AppData\Roaming\Python\Python37\Scripts`
- MSIprogramma di installazione (64 bit) — `C:\Program Files\Amazon\AWSCLI\bin`
- MSIprogramma di installazione (32 bit) — `C:\Program Files (x86)\Amazon\AWSCLI\bin`

Attenersi alla seguente procedura se viene restituito un percorso di file.

A file path is returned

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

Puoi trovare la posizione di installazione del programma `aws` con il comando seguente.

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

A file path is NOT returned

Se il comando `where` restituisce il seguente errore, non è nel PATH del sistema e non è possibile eseguirlo digitandone il nome.

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

```
INFO: Could not find files for the given pattern(s).
```

In questo caso, eseguire il comando `where` con il parametro `/R path` per specificare di ricercare in tutte le cartelle, quindi aggiungere il percorso manualmente. Utilizzare la riga di comando o Esplora file per scoprire dove è installato sul computer.

```
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. Premere il tasto Windows e immettere **environment variables**.
3. Scegli Modifica variabili di ambiente per l'account.
4. Scegli PATH, quindi scegli Modifica.
5. Aggiungere il percorso trovato nel campo Valore della variabile, ad esempio, **C:\Program Files\Amazon\AWSCLI\bin\aws.exe**.
6. Fai doppio clic su OK per applicare le nuove impostazioni.
7. Chiudi tutte le richieste di comando in esecuzione e riapri la finestra del prompt dei comandi.

## Risoluzione degli errori di AWS CLI installazione e disinstallazione

Se riscontri problemi dopo l'installazione o la disinstallazione di AWS CLI, consulta la procedura [Risolvere gli errori](#) per la risoluzione dei problemi. Per le procedure di risoluzione dei problemi più importanti, consulta [the section called "Errori relativi al comando non trovato"](#), [the section called "Il comando `aws --version` restituisce una versione diversa da quella installata"](#), e [the section called "Il comando `aws --version` restituisce una versione dopo la disinstallazione di AWS CLI"](#).

## Installazione e aggiornamento della AWS CLI versione 1 in un ambiente virtuale

È possibile evitare conflitti tra le versioni dei requisiti e gli altri pip pacchetti installando la versione 1 di AWS Command Line Interface (AWS CLI) in un ambiente virtuale.

### Argomenti

- [Prerequisiti](#)
- [Installa e aggiorna la AWS CLI versione 1 in un ambiente virtuale](#)
- [Risoluzione degli errori di AWS CLI installazione e disinstallazione](#)

## Prerequisiti

- Python 3.8 o successivo. Per istruzioni sull'installazione, consulta la pagina del [download di Python](#) nel manuale Beginner Guidedi Python.

Matrice di supporto per la versione Python

AWS CLI versione	Versione Python supportata
1.32.0 — attuale	Python 3.8+
1.27.0 — 1,31.x	Python 3.7+
1.20.0 — 1,26.x	Python 3.6+
1.19.0 — 1.19x	Python 2.7+, Python 3.6+
1.17 — 1.18x	Python 2.7+, Python 3.4+
1,0 — 1,16x	Python 2.6 e versioni precedenti, Python 3.3 e precedenti

- È installato pip o pip3.

## Installa e aggiorna la AWS CLI versione 1 in un ambiente virtuale

1. Installare virtualenv utilizzando pip.

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

2. Creare un ambiente virtuale e assegnargli un nome.

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

In alternativa, è possibile utilizzare l'opzione `-p` per specificare una versione di Python diversa da quella di default.

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

3. Attivare il nuovo ambiente virtuale.

Linux o macOS

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

Windows

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

Il prompt cambia per mostrare che l'ambiente virtuale è attivo:

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

4. Installa o aggiorna la AWS CLI versione 1 nel tuo ambiente virtuale.

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

5. Verifica che la AWS CLI versione 1 sia installata correttamente.

```
$ 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. Puoi utilizzare il comando `deactivate` per chiudere l'ambiente virtuale. Quando avvii una nuova sessione, devi attivare nuovamente l'ambiente.

## Risoluzione degli errori di AWS CLI installazione e disinstallazione

Se riscontri problemi dopo l'installazione o la disinstallazione di AWS CLI, consulta la procedura [Risolvere gli errori](#) per la risoluzione dei problemi. Per le procedure di risoluzione dei problemi più importanti, consulta [the section called "Errori relativi al comando non trovato"](#), [the section called "Il comando `aws --version` restituisce una versione diversa da quella installata"](#), e [the section called "Il comando `aws --version` restituisce una versione dopo la disinstallazione di AWS CLI"](#).

# Configurazione delle impostazioni per AWS CLI

Questa sezione spiega come configurare le impostazioni con cui AWS Command Line Interface (AWS CLI) interagisce AWS. Questi sono i seguenti:

- Le credenziali identificano chi sta chiamando il API. Le credenziali di accesso vengono utilizzate per crittografare la richiesta ai AWS server per confermare l'identità e recuperare le politiche di autorizzazione associate. Queste autorizzazioni determinano le azioni che è possibile eseguire. Per informazioni sulla configurazione delle credenziali, consulta. [Credenziali di autenticazione e accesso](#)
- Altri dettagli di configurazione per indicare AWS CLI come elaborare le richieste, come il formato di output predefinito e la AWS regione predefiniti.

## Note

AWS richiede che tutte le richieste in arrivo siano firmate crittograficamente. Lo AWS CLI fa per te. La «firma» include un 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 documento riconosciuto dal AWS servizio, AWS respinge la richiesta.

## Configurazione e precedenza delle credenziali

Le credenziali e le impostazioni di configurazione si trovano in più posizioni, ad esempio nelle variabili di sistema o di ambiente utente, nei file di AWS configurazione locali, oppure sono dichiarate esplicitamente nella riga di comando come parametro. Alcune posizioni hanno la precedenza su altre. Le AWS CLI credenziali e le impostazioni di configurazione hanno la precedenza nell'ordine seguente:

1. [Opzioni della riga di comando](#): sostituisce le impostazioni in qualsiasi altra posizione, ad esempio i parametri `--region`, `--output` e `--profile`
2. [Variabili di ambiente](#): è possibile memorizzare valori nelle variabili di ambiente del sistema.
3. [Assumi un ruolo](#): assumi le autorizzazioni di un IAM ruolo tramite la configurazione o il [aws sts assume-role](#) comando.

4. [Assumi il ruolo con l'identità web](#): assumi le autorizzazioni di un IAM ruolo utilizzando l'identità web tramite la configurazione o il [aws sts assume-role](#) comando.
5. [File delle credenziali](#): il config file `credentials` and viene aggiornato quando si esegue il comando `aws configure`. Il `credentials` file si trova `~/.aws/credentials` in Linux o macOS o `C:\Users\USERNAME\.aws\credentials` in Windows.
6. [Processo personalizzato](#): ottieni le tue credenziali da una fonte esterna.
7. [File di configurazione](#): il config file `credentials` and viene aggiornato quando si esegue il comando `aws configure`. Il config file si trova `~/.aws/config` in Linux o macOS o `C:\Users\USERNAME\.aws\config` in Windows.
8. [Credenziali del contenitore](#): puoi associare un IAM ruolo a ciascuna delle definizioni di attività di Amazon Elastic Container Service (Amazon ECS). Le credenziali temporanee per tale ruolo sono quindi disponibili per i contenitori dell'attività. Per ulteriori informazioni, consulta [IAMRoles for Tasks](#) nella Amazon Elastic Container Service Developer Guide.
9. [Credenziali del profilo dell'EC2 istanza Amazon](#): puoi associare un IAM ruolo a ciascuna delle tue istanze Amazon Elastic Compute Cloud EC2 (Amazon). Le credenziali temporanee per tale ruolo sono quindi disponibili per il codice in esecuzione nell'istanza. Le credenziali vengono fornite tramite il servizio di EC2 metadati Amazon. Per ulteriori informazioni, consulta [IAMRoles for Amazon EC2 nella Amazon EC2 User Guide](#) e [Using Instance Profiles](#) nella IAM User Guide.

## Argomenti aggiuntivi in questa sezione

- [the section called “Impostazioni dei file di configurazione e credenziali in AWS CLI”](#)
- [the section called “Variabili di ambiente”](#)
- [the section called “Opzioni della riga di comando in AWS CLI”](#)
- [the section called “Configurazione del completamento dei comandi in AWS CLI”](#)
- [the section called “Tentativi”](#)
- [the section called “Utilizzo di un HTTP proxy per AWS CLI”](#)

## Impostazioni dei file di configurazione e credenziali in AWS CLI

Puoi salvare le impostazioni di configurazione e le credenziali utilizzate di frequente nei file gestiti da AWS CLI.



I file sono suddivisi in `profiles`. Per impostazione predefinita, AWS CLI utilizza le impostazioni presenti nel profilo denominato `default`. Per usare impostazioni alternative, puoi creare profili aggiuntivi e farvi riferimento.

Puoi sostituire una singola impostazione configurando una delle variabili di ambiente supportate oppure utilizzando un parametro della riga di comando. Per ulteriori informazioni sulla precedenza delle impostazioni di configurazione, consulta [Configurazione delle impostazioni per AWS CLI](#).

#### Note

Per informazioni sulla configurazione delle credenziali, consulta [Credenziali di autenticazione e accesso](#).

## Argomenti

- [Formato dei file di configurazione e delle credenziali](#)
- [Dove vengono archiviate le impostazioni di configurazione?](#)
- [Utilizzo di profili denominati](#)
- [Imposta e visualizza le impostazioni di configurazione utilizzando i comandi](#)
- [Impostazione di nuovi esempi di comandi per la configurazione e le credenziali](#)
- [Impostazioni supportate del file config](#)

## Formato dei file di configurazione e delle credenziali

I `credentials` file `config` e sono organizzati in sezioni. Le sezioni includono profili e servizi. Una sezione è una raccolta denominata di impostazioni e continua fino a quando non viene incontrata un'altra riga di definizione della sezione. È possibile memorizzare più profili e sezioni nei `credentials` file `config` and.

Questi file sono file di testo semplice che utilizzano il seguente formato:

- I nomi delle sezioni sono racchiusi tra parentesi `[]` come, `e[default]`. `[profile user1] [sso-session]`
- Tutte le voci di una sezione assumono il formato generale di `setting_name=value`
- Le righe possono essere commentate iniziando la riga con un carattere cancelletto `()#`.





```
region=us-east-1
output=text
```

## Amazon EC2 instance metadata credentials

Questo esempio riguarda le credenziali ottenute dai metadati dell'EC2istanza Amazon di hosting. Per ulteriori informazioni, consulta [the section called “Utilizzo dei metadati delle EC2 istanze Amazon come credenziali nel AWS CLI”](#).

### File di credenziali

Il `credentials` file non viene utilizzato per questo metodo di autenticazione.

### File di configurazione

```
[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

Per evitare rischi per la sicurezza, non utilizzate IAM gli utenti per l'autenticazione quando sviluppate software appositamente creato o lavorate con dati reali. Utilizza invece la federazione con un provider di identità come [AWS IAM Identity Center](#).

Questo esempio riguarda le credenziali a lungo termine di AWS Identity and Access Management. Per ulteriori informazioni, consulta [the section called “IAMutenti”](#).

### File di credenziali

```
[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
```

## File di configurazione

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

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

Per ulteriori informazioni e metodi di autorizzazione e credenziali aggiuntivi, vedere. [the section called “IAMutenti”](#)

## Tipo di sezione: **services**

La `services` sezione è un gruppo di impostazioni che configura gli endpoint personalizzati per Servizio AWS le richieste. Un profilo viene quindi collegato a una `services` sezione.

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

La `services` sezione è separata in sottosezioni da `<SERVICE> =` linee, dove si `<SERVICE>` trova la chiave Servizio AWS identificativa. L' Servizio AWS identificatore si basa su quello del API modello `serviceId` sostituendo tutti gli spazi con caratteri di sottolineatura e tutte le lettere minuscole. Per un elenco di tutte le chiavi identificative dei servizi da utilizzare nella sezione, vedere. [services Utilizzo degli endpoint in AWS CLI](#) La chiave identificativa del servizio è seguita da impostazioni annidate, ciascuna sulla propria riga e rientrata da due spazi.

L'esempio seguente configura l'endpoint da utilizzare per le richieste effettuate al servizio in Amazon DynamoDB `my-services` sezione utilizzata in `dev` profilo. Tutte le righe immediatamente successive che sono rientrate sono incluse in tale sottosezione e si applicano a tale servizio.

```
[profile dev]
```

```
services = my-services

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

Per ulteriori informazioni sugli endpoint specifici del servizio, vedere. [Utilizzo degli endpoint in AWS CLI](#)

Se il tuo profilo ha credenziali basate sui ruoli configurate tramite un `source_profile` parametro per IAM assumere la funzionalità del ruolo, utilizza SDK solo le configurazioni di servizio per il profilo specificato. Non utilizza profili concatenati a ruoli. Ad esempio, utilizzando il seguente config file condiviso:

```
[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
```

Se usi il profilo B ed effettui una chiamata nel tuo codice verso AmazonEC2, l'endpoint si risolve come. `https://profile-b-ec2-endpoint.aws` Se il codice invia una richiesta a qualsiasi altro servizio, la risoluzione dell'endpoint non seguirà alcuna logica personalizzata. L'endpoint non si risolve nell'endpoint globale definito nel profilo. A Affinché un endpoint globale abbia effetto sul profiloB, è necessario `endpoint_url` impostarlo direttamente all'interno del profilo. B

## Dove vengono archiviate le impostazioni di configurazione?

AWS CLI Memorizza le informazioni riservate sulle credenziali specificate `aws configure` in un file locale denominato `credentials`, in una cartella denominata `.aws` nella home directory. Le altre opzioni di configurazione meno riservate specificate con `aws configure` vengono archiviate in un file locale denominato `config` che è archiviato nella cartella `.aws` della home directory.

### Archiviazione delle credenziali nel file di configurazione

Puoi conservare tutte le impostazioni del tuo profilo in un unico file in quanto AWS CLI possono leggere le credenziali dal `config` file. Se in entrambi i file sono presenti credenziali per un profilo che condivide lo stesso nome, le chiavi nel file delle credenziali hanno la precedenza. Ti consigliamo di conservare le credenziali nei file `credentials`. Questi file vengono utilizzati anche dai kit di sviluppo software in vari linguaggi (SDKs). Se utilizzi uno di questi SDKs in aggiunta a AWS CLI, conferma se le credenziali devono essere archiviate nel loro file.

La posizione della home directory varia in base al sistema operativo, ma viene definita utilizzando le variabili di ambiente `%UserProfile%` in Windows e `$HOME` o `~` (tilde) nei sistemi basati su Unix. È possibile specificare una posizione non predefinita per i file impostando le variabili di ambiente `AWS_SHARED_CREDENTIALS_FILE` e `AWS_CONFIG_FILE` e su un altro percorso locale. Per informazioni dettagliate, vedi [Configurazione delle variabili di ambiente per AWS CLI](#).

Quando utilizzate un profilo condiviso che specifica un ruolo AWS Identity and Access Management (IAM), AWS CLI chiama l'operazione `AWS STS AssumeRole` per recuperare le credenziali temporanee. Queste credenziali vengono quindi archiviate (in `~/.aws/cli/cache`). AWS CLI I comandi successivi utilizzano le credenziali temporanee memorizzate nella cache fino alla loro scadenza e, a quel punto, le aggiornano automaticamente. AWS CLI

## Utilizzo di profili denominati

Se nessun profilo è definito in modo esplicito, viene utilizzato il profilo `default`.

Per usare un profilo denominato, aggiungi l'opzione `--profile profile-name` al comando. L'esempio seguente elenca tutte le tue EC2 istanze Amazon utilizzando le credenziali e le impostazioni definite nel `user1` profilo.

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

Per utilizzare un profilo denominato per più comandi, puoi evitare di specificare il profilo in ogni comando impostando la variabile di ambiente `AWS_PROFILE` come profilo predefinito. È possibile ignorare questa impostazione utilizzando il `--profile` parametro.

## Linux or macOS

```
$ export AWS_PROFILE=user1
```

## Windows

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

Se si utilizza [set](#) per impostare una variabile di ambiente, il valore utilizzato viene modificato fino al termine della sessione del prompt dei comandi corrente o finché non imposti la variabile su un valore diverso.

L'uso di [setx](#) per impostare una variabile di ambiente comporta la modifica del valore in tutte le shell di comando create dopo l'esecuzione del comando. Non influisce su alcuna shell di comando già in esecuzione nel momento in cui si esegue il comando. Chiudere e riavviare la shell dei comandi per visualizzare gli effetti della modifica.

L'impostazione della variabile di ambiente modifica il profilo di default fino al termine della sessione della shell o finché non si imposta la variabile su un valore diverso. Puoi rendere le variabili di ambiente persistenti per le sessioni future inserendole nello script di avvio della shell. Per ulteriori informazioni, consulta [Configurazione delle variabili di ambiente per AWS CLI](#).

## Imposta e visualizza le impostazioni di configurazione utilizzando i comandi

Esistono diversi modi per visualizzare e configurare le impostazioni di configurazione utilizzando i comandi.

### [aws configure](#)

Esegui questo comando per impostare e visualizzare rapidamente le credenziali, la regione e il formato di output. L'esempio seguente illustra i valori di esempio.

```
$ 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

Puoi impostare le credenziali o le impostazioni di configurazione utilizzando `aws configure set`. Specifica il profilo che intendi visualizzare o modificare con l'impostazione `--profile`.

Ad esempio, il comando seguente configura l'impostazione `region` nel profilo denominato `integ`.

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

Per rimuovere un'impostazione, elimina manualmente l'impostazione nei `credentials file config` e in un editor di testo.

## aws configure get

Puoi recuperare le credenziali o le impostazioni di configurazione impostate utilizzando `aws configure get`. Specifica il profilo che intendi visualizzare o modificare con l'impostazione `--profile`.

Ad esempio, il comando seguente recupera l'impostazione `region` nel profilo denominato `integ`.

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

Se l'output è vuoto, l'impostazione non è configurata esplicitamente e viene utilizzato il valore predefinito.

## aws configure list

Per elencare i dati di configurazione, utilizzare il `aws configure list` comando. Questo comando elenca le informazioni di configurazione del profilo, della chiave di accesso, della chiave segreta e della regione utilizzate per il profilo specificato. Per ogni elemento di configurazione, mostra il valore, dove è stato recuperato il valore di configurazione e il nome della variabile di configurazione.

Ad esempio, se fornite il Regione AWS in una variabile di ambiente, questo comando mostra il nome dell'area che avete configurato, che questo valore proviene da una variabile di ambiente e il nome della variabile di ambiente.

Per i metodi di credenziali temporanei come `roles` e IAM Identity Center, questo comando visualizza la chiave di accesso temporaneamente memorizzata nella cache e viene visualizzata la chiave di accesso segreta.



```
$ aws configure set source_profile default
$ aws configure set role_session_name session_user1
$ aws configure set region us-west-2
$ aws configure set output json
```

## Amazon EC2 instance metadata credentials

Questo esempio riguarda le credenziali ottenute dai metadati dell'EC2istanza Amazon di hosting. Non esiste una procedura guidata per questo processo, pertanto ogni valore viene impostato utilizzando il comando `aws configure set`. Per ulteriori informazioni, consulta [the section called “Utilizzo dei metadati delle EC2 istanze Amazon come credenziali nel 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

Per evitare rischi per la sicurezza, non utilizzate IAM gli utenti per l'autenticazione quando sviluppate software appositamente creato o lavorate con dati reali. Utilizza invece la federazione con un provider di identità come [AWS IAM Identity Center](#).

Questo esempio riguarda le credenziali a lungo termine di AWS Identity and Access Management. Per ulteriori informazioni, consulta [the section called “IAM utenti”](#).

```
$ 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
```

## Impostazioni supportate del file `config`

### Argomenti

- [Impostazioni globali](#)

- [Impostazioni dei comandi S3 personalizzati](#)

Le seguenti impostazioni sono supportate nel file `config`. Vengono utilizzati i valori elencati nel profilo specificato (o nel profilo predefinito), a meno che non vengano sostituiti grazie alla presenza di una variabile di ambiente con lo stesso nome o di un'opzione della riga di comando con lo stesso nome. Per ulteriori informazioni sulle impostazioni di ordine che hanno la precedenza, consulta [Configurazione delle impostazioni per AWS CLI](#)

## Impostazioni globali

### *api\_versions*

Alcuni AWS servizi mantengono più API versioni per supportare la compatibilità con le versioni precedenti. Per impostazione predefinita, AWS CLI i comandi utilizzano l'ultima API versione disponibile. È possibile specificare una API versione da utilizzare per un profilo includendo l'`api_versions` impostazione nel `config` file.

Si tratta di un'impostazione «annidata» seguita da una o più righe rientrate, ciascuna delle quali identifica un AWS servizio e la API versione da utilizzare. Consulta la documentazione di ogni servizio per capire quali API versioni sono disponibili.

L'esempio seguente mostra come specificare una API versione per due AWS servizi. Queste API versioni vengono utilizzate solo per i comandi eseguiti con il profilo che contiene queste impostazioni.

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

Questa impostazione non dispone di una variabile di ambiente o di un parametro della riga di comando equivalente.

### **aws\_access\_key\_id**

Specifica la chiave di AWS accesso utilizzata come parte delle credenziali per autenticare la richiesta di comando. Sebbene tale impostazione possa essere archiviata nel file `config`, ti consigliamo di archivarla nel file `credentials`.

Può essere sostituita dalla variabile di ambiente `AWS_ACCESS_KEY_ID`. Non è possibile specificare l'ID chiave di accesso come un'opzione della riga di comando.

```
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
```

## `aws_secret_access_key`

Specifica la chiave AWS segreta utilizzata come parte delle credenziali per autenticare la richiesta di comando. Sebbene tale impostazione possa essere archiviata nel file `config`, ti consigliamo di archivarla nel file `credentials`.

Può essere sostituita dalla variabile di ambiente `AWS_SECRET_ACCESS_KEY`. Non è possibile specificare la chiave di accesso segreta come un'opzione della riga di comando.

```
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

## `aws_session_token`

Specifica un token di sessione. AWS È richiesto un token di sessione solo se si specificano manualmente credenziali di sicurezza temporanee. Sebbene tale impostazione possa essere archiviata nel file `config`, ti consigliamo di archivarla nel file `credentials`.

Può essere sostituita dalla variabile di ambiente `AWS_SESSION_TOKEN`. Non è possibile specificare il token di sessione come un'opzione della riga di comando.

```
aws_session_token = AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT  
+FvwnKwRc0IfrrRh3c/LTo6UDdyJw00vEVPvLXCrrrUtdnniCEXAMPLE/  
IvU1dYUg2RVAJBanLiHb4IgrmpRV3zrkuWJ0gQs8IZZaIv2BXIa2R40lgk
```

## `ca_bundle`

Specifica un pacchetto di certificati CA (un file con `.pem` estensione) utilizzato per verificare i certificati. SSL

Può essere sostituita dalla variabile di ambiente [`AWS\_CA\_BUNDLE`](#) o dall'opzione della riga di comando [`--ca-bundle`](#).

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

## cli\_follow\_urlparam

Specifica se i AWS CLI tentativi di seguire i URL collegamenti nei parametri della riga di comando che iniziano con `http://` o `https://`. Se abilitato, il contenuto recuperato viene utilizzato come valore del parametro anziché URL.

- `true`: questo è il valore predefinito. Se specificato, vengono recuperati i parametri di stringa che iniziano con `http://` o `https://` e i contenuti scaricati vengono utilizzati come valore di parametro per il comando.
- `false` — Se specificato, AWS CLI non tratta i valori delle stringhe dei parametri che iniziano con `http://` o `https://` in modo diverso dalle altre stringhe.

Questa voce non dispone di una variabile di ambiente o di un'opzione della riga di comando equivalente.

```
cli_follow_urlparam = false
```

## cli\_history

Disabilitato per impostazione predefinita. Questa impostazione abilita la cronologia dei comandi per AWS CLI. Dopo aver abilitato questa impostazione, AWS CLI registra la cronologia dei comandi.

```
cli_history = enabled
```

È possibile elencare la cronologia utilizzando il `aws history list` comando e utilizzare il risultato `command_ids` del `aws history show` comando per i dettagli. Per ulteriori informazioni, [aws history](#) consulta la guida AWS CLI di riferimento.

## cli\_timestamp\_format

Specifica il formato dei valori di timestamp inclusi nell'output. Puoi specificare uno dei seguenti valori:

- `iso8601` — Il valore predefinito per la versione 2. AWS CLI [Se specificato, AWS CLI riformatta tutti i timestamp in base a 8601. ISO](#)

ISO1 timestamp formattati 8601 sono simili agli esempi seguenti. Il primo esempio mostra l'ora in [Coordinated Universal Time \(UTC\)](#) includendo un Z valore dopo l'ora. La data e l'ora sono separati da una T.

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

Per specificare un fuso orario diverso, anziché ilZ, specifica un + o - e il numero di ore in cui il fuso orario desiderato è anteriore o posterioreUTC, come valore a due cifre. L'esempio seguente mostra la stessa ora dell'esempio precedente ma adattata all'ora solare del Pacifico, che è indietro di otto ore. UTC

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

- `wire` — Il valore predefinito per la AWS CLI versione 1. Se specificato, AWS CLI visualizza tutti i valori del timestamp esattamente come ricevuti nella risposta alla HTTP query.

Questa voce non dispone di una variabile di ambiente o di un'opzione della riga di comando equivalente.

```
cli_timestamp_format = iso8601
```

## credential\_process

Specifica un comando esterno da eseguire per generare o recuperare le AWS CLI credenziali di autenticazione da utilizzare per questo comando. Il comando deve restituire le credenziali in un formato specifico. Per ulteriori informazioni sull'utilizzo di questa impostazione, consulta [Acquisizione di credenziali con un processo esterno in AWS CLI](#).

Questa voce non dispone di una variabile di ambiente o di un'opzione della riga di comando equivalente.

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

## credential\_source

Utilizzato all'interno di EC2 istanze o contenitori Amazon per specificare dove AWS CLI trovare le credenziali da utilizzare per assumere il ruolo specificato con il `role_arn` parametro. Non è possibile specificare sia `source_profile` sia `credential_source` nello stesso profilo.

Questo parametro può avere uno dei seguenti tre valori:

- `Ambiente`: specifica che deve recuperare le AWS CLI credenziali di origine dalle variabili di ambiente.

- `Ec2InstanceMetadata`: specifica che deve essere utilizzato il IAM ruolo AWS CLI associato al [profilo dell'EC2istanza](#) per ottenere le credenziali di origine.
- `EcsContainer`— Specifica che deve essere utilizzato AWS CLI il IAM ruolo associato al ECS contenitore come credenziali di origine.

```
credential_source = Ec2InstanceMetadata
```

## `duration_seconds`

Specifica la durata massima della sessione del ruolo, in secondi. Questo valore può variare da 900 secondi (15 minuti) fino alla durata massima della sessione per il ruolo (massimo 43200 minuti). Si tratta di un parametro facoltativo e, per impostazione predefinita, il valore è impostato su 3600 secondi.

## `endpoint_url`

Specifica l'endpoint utilizzato per tutte le richieste di servizio. Se questa impostazione viene utilizzata nella [services](#) sezione del config file, l'endpoint viene utilizzato solo per il servizio specificato.

L'esempio seguente utilizza l'endpoint globale `http://localhost:1234` e un endpoint specifico del servizio di per Amazon `http://localhost:4567 S3`.

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

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

Le impostazioni di configurazione degli endpoint si trovano in più posizioni, ad esempio le variabili di sistema o di ambiente utente, i file di AWS configurazione locali o sono dichiarate esplicitamente nella riga di comando come parametro. Le impostazioni di configurazione AWS CLI degli endpoint hanno la precedenza nell'ordine seguente:

1. L'opzione della [--endpoint-url](#) riga di comando.
2. Se abilitata, la variabile di ambiente [AWS\\_IGNORE\\_CONFIGURED\\_ENDPOINT\\_URLS](#) globale dell'endpoint o l'impostazione del profilo [ignore\\_configure\\_endpoint\\_urls](#) per ignorare gli endpoint personalizzati.



3. Il valore fornito da una variabile [AWS\\_ENDPOINT\\_URL\\_<SERVICE>](#) di ambiente specifica del servizio, ad esempio. `AWS_ENDPOINT_URL_DYNAMODB`
4. I valori forniti dalle variabili [AWS\\_USE\\_DUALSTACK\\_ENDPOINT](#), [AWS\\_USE\\_FIPS\\_ENDPOINT](#), e di [AWS\\_ENDPOINT\\_URL](#) ambiente.
5. Il valore dell'endpoint specifico del servizio fornito dall'[endpoint\\_url](#) impostazione all'interno di una `services` sezione del file condiviso. `config`
6. Il valore fornito dall'[endpoint\\_url](#) impostazione all'interno di uno `profile` dei file condivisi. `config`
7. [use\\_dualstack\\_endpoint](#), [use\\_fips\\_endpoint](#) e [endpoint\\_url](#) impostazioni.
8. Qualsiasi endpoint predefinito URL per il rispettivo dispositivo Servizio AWS viene utilizzato per ultimo. Per un elenco degli endpoint del servizio standard disponibili in ogni regione, consulta [Regioni ed endpoint AWS](#) nella Riferimenti generali di Amazon Web Services.

### **ignore\_configure\_endpoint\_urls**

Se abilitato, AWS CLI ignora tutte le configurazioni personalizzate degli endpoint specificate nel file. `config` I valori validi sono **true** e **false**.

```
ignore_configure_endpoint_urls = true
```

Le impostazioni di configurazione degli endpoint si trovano in più posizioni, ad esempio le variabili di sistema o di ambiente utente, i file di AWS configurazione locali o sono dichiarate esplicitamente nella riga di comando come parametro. Le impostazioni di configurazione AWS CLI degli endpoint hanno la precedenza nell'ordine seguente:

1. L'opzione della [--endpoint-url](#) riga di comando.
2. Se abilitata, la variabile di ambiente [AWS\\_IGNORE\\_CONFIGURED\\_ENDPOINT\\_URLS](#) globale dell'endpoint o l'impostazione del profilo [ignore\\_configure\\_endpoint\\_urls](#) per ignorare gli endpoint personalizzati.
3. Il valore fornito da una variabile [AWS\\_ENDPOINT\\_URL\\_<SERVICE>](#) di ambiente specifica del servizio, ad esempio. `AWS_ENDPOINT_URL_DYNAMODB`
4. I valori forniti dalle variabili [AWS\\_USE\\_DUALSTACK\\_ENDPOINT](#), [AWS\\_USE\\_FIPS\\_ENDPOINT](#), e di [AWS\\_ENDPOINT\\_URL](#) ambiente.
5. Il valore dell'endpoint specifico del servizio fornito dall'[endpoint\\_url](#) impostazione all'interno di una `services` sezione del file condiviso. `config`
6. Il valore fornito dall'[endpoint\\_url](#) impostazione all'interno di uno `profile` dei file condivisi. `config`

7. [use\\_dualstack\\_endpoint](#), [use\\_fips\\_endpoint](#), [endpoint\\_url](#) impostazioni.
8. Qualsiasi endpoint predefinito URL per il rispettivo dispositivo Servizio AWS viene utilizzato per ultimo. Per un elenco degli endpoint del servizio standard disponibili in ogni regione, consulta [Regioni ed endpoint AWS](#) nella Riferimenti generali di Amazon Web Services.

## [external\\_id](#)

Specifica un identificatore univoco che viene utilizzato da terze parti per assumere un ruolo negli account dei relativi clienti. Questo è mappato al parametro `ExternalId` nell'operazione `AssumeRole`. Questo parametro è necessario solo se la policy di attendibilità per il ruolo specifica un valore per `ExternalId`. Per ulteriori informazioni, consulta [Come utilizzare un ID esterno per concedere l'accesso alle AWS risorse a terzi](#) nella Guida per l'IAM utente.

## [max\\_attempts](#)

Specificate il valore massimo di tentativi utilizzato dal gestore dei tentativi, in AWS CLI base al quale la chiamata iniziale viene conteggiata ai fini del `max_attempts` valore fornito.

È possibile sovrascrivere questo valore utilizzando la variabile di ambiente. `AWS_MAX_ATTEMPTS`

```
max_attempts = 3
```

## [mfa\\_serial](#)

Il numero di identificazione di un MFA dispositivo da utilizzare quando si assume un ruolo. Questo è obbligatorio solo se la politica di fiducia del ruolo assunto include una condizione che richiede MFA l'autenticazione. Il valore può essere un numero di serie per un dispositivo hardware (come `GAHT12345678`) o un Amazon Resource Name (ARN) per un MFA dispositivo virtuale (come `arn:aws:iam::123456789012:mfa/user`).

## output

Specifica il formato di output predefinito per i comandi richiesti utilizzando questo profilo. Puoi specificare uno qualsiasi dei valori seguenti:

- **[json](#)**— L'output è formattato come [JSON](#) stringa.
- **[text](#)**— L'output è formattato come più righe di valori di stringa separati da tabulazioni. Questo può essere utile per passare l'output a un elaboratore di testi, ad esempio `grep`, `sed` o `awk`.
- **[table](#)**— L'output viene formattato come tabella utilizzando i caratteri `+|-` per formare i bordi delle celle. In genere presenta le informazioni in un formato comprensibile molto più semplice da leggere rispetto ad altri, ma non altrettanto utile a livello programmatico.

Può essere sostituita dalla variabile di ambiente `AWS_DEFAULT_OUTPUT` o dall'opzione della riga di comando `--output`.

```
output = table
```

## **parameter\_validation**

Specifica se il AWS CLI client tenta di convalidare i parametri prima di inviarli all'endpoint del servizio. AWS

- `true` — Questo è il valore predefinito. Se specificato, AWS CLI esegue la convalida locale dei parametri della riga di comando.
- `false` — Se specificato, AWS CLI non convalida i parametri della riga di comando prima di inviarli all'endpoint del AWS servizio.

Questa voce non dispone di una variabile di ambiente o di un'opzione della riga di comando equivalente.

```
parameter_validation = false
```

## **region**

Specifica a chi Regione AWS inviare le richieste per i comandi richiesti utilizzando questo profilo.

- È possibile specificare uno qualsiasi dei codici regionali disponibili per il servizio scelto, elencati in [AWS Regioni ed endpoint](#) in. Riferimenti generali di Amazon Web Services
- `aws_global` consente di specificare l'endpoint globale per i servizi che supportano un endpoint globale oltre agli endpoint regionali, come AWS Security Token Service (AWS STS) e Amazon Simple Storage Service (Amazon S3).

Puoi sovrascrivere questo valore utilizzando la variabile di ambiente o `AWS_DEFAULT_REGION` l'opzione della riga di comando. `--region`

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

## **retry\_mode**

Specificate la modalità di riprova utilizzata. AWS CLI Sono disponibili tre modalità di riprova: `legacy` (predefinita), `standard` e `adattiva`. Per ulteriori informazioni sui tentativi, consultare [AWS CLI nuovi tentativi nel AWS CLI](#).

È possibile sovrascrivere questo valore utilizzando la `AWS_RETRY_MODE` variabile di ambiente.

```
retry_mode = standard
```

## role\_arn

Specifica l'Amazon Resource Name (ARN) di un IAM ruolo che desideri utilizzare per eseguire i CLI comandi. Puoi anche specificare uno dei seguenti parametri per identificare le credenziali che dispongono dell'autorizzazione per assumere questo ruolo:

- `source_profile`
- `credential_source`

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

La variabile di ambiente [AWS\\_ROLE\\_ARN](#) sostituisce questa impostazione.

Per ulteriori informazioni sull'utilizzo delle identità Web, vedere [the section called “Assunzione di un ruolo con un'identità Web”](#)

## role\_session\_name

Specifica il nome da associare alla sessione del ruolo. Questo valore viene fornito al `RoleSessionName` parametro quando AWS CLI chiama l'`AssumeRole` operazione e diventa parte del ruolo assunto userARN: `arn:aws:sts::123456789012:assumed-role/role_name/role_session_name`. Si tratta di un parametro facoltativo. Se non fornisci questo valore, viene automaticamente generato un nome della sessione. Questo nome viene visualizzato nei log di AWS CloudTrail per le voci associate a questa sessione.

```
role_session_name = maria_garcia_role
```

La variabile di ambiente ha la [AWS\\_ROLE\\_SESSION\\_NAME](#) precedenza su questa impostazione.

Per ulteriori informazioni sull'utilizzo delle identità Web, vedere [the section called “Assunzione di un ruolo con un'identità Web”](#)

## services

Specifica la configurazione del servizio da utilizzare per il profilo.

```
[profile dev-s3-specific-and-global]
```

```
endpoint_url = http://localhost:1234
services = s3-specific

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

Per ulteriori informazioni sulla `services` sezione, vedere [the section called “services”](#).

La variabile di ambiente ha la [AWS\\_ROLE\\_SESSION\\_NAME](#) precedenza su questa impostazione.

Per ulteriori informazioni sull'utilizzo delle identità Web, vedere. [the section called “Assunzione di un ruolo con un'identità Web”](#)

## **sdk\_ua\_app\_id**

Una singola Account AWS può essere utilizzata da più applicazioni del cliente a cui effettuare Servizi AWS chiamate. L'ID dell'applicazione identifica quale applicazione di origine ha effettuato una serie di chiamate utilizzando un Servizio AWS. AWS SDK e i servizi non utilizzano o interpretano questo valore se non per riemergere nelle comunicazioni con i clienti. Ad esempio, questo valore può essere incluso nelle e-mail operative per identificare in modo univoco quale delle applicazioni è associata alla notifica.

L'ID dell'applicazione è una stringa con una lunghezza massima di 50 caratteri. Sono consentite lettere, numeri e i seguenti caratteri speciali: Per ! \$ % & \* + - . , ^ \_ ` | ~ impostazione predefinita, non viene assegnato alcun valore.

```
sdk_ua_app_id = prod1
```

Questa impostazione può essere sovrascritta utilizzando la variabile di ambiente [AWS\\_SDK\\_UA\\_APP\\_ID](#). Non è possibile impostare questo valore come parametro della riga di comando.

## **source\_profile**

Specifica un profilo denominato con credenziali a lungo termine che l' AWS CLI può utilizzare per assumere un ruolo specificato con il parametro `role_arn`. Non è possibile specificare sia `source_profile` sia `credential_source` nello stesso profilo.

```
source_profile = production-profile
```

## *sts\_regional\_endpoints*

Specifica in che modo AWS CLI determina l'endpoint del AWS servizio utilizzato AWS CLI dal client per parlare con AWS Security Token Service (AWS STS). Il valore predefinito per la AWS CLI versione 1 è `legacy`.

È possibile specificare uno dei due valori:

- **legacy**— Utilizza l'endpoint globale `sts.amazonaws.com`, per le seguenti AWS regioni: `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`, `us-west-2`. Tutte le altre regioni utilizzano automaticamente il rispettivo endpoint regionale.
- **regional**— Utilizza AWS CLI sempre l'AWS STS endpoint per la regione attualmente configurata. Ad esempio, se il client è configurato per `us-west-2`, tutte le chiamate a AWS STS vengono effettuate all'endpoint regionale `sts.us-west-2.amazonaws.com` anziché all'endpoint globale `sts.amazonaws.com`. Per inviare una richiesta all'endpoint globale mentre questa impostazione è abilitata, è possibile impostare l'area geografica su `aws-global`.

Questa impostazione può essere sovrascritta utilizzando la variabile di ambiente `AWS_STS_REGIONAL_ENDPOINTS`. Non è possibile impostare questo valore come parametro della riga di comando.

## *use\_dualstack\_endpoint*

Abilita l'uso di endpoint dual-stack per inviare richieste. AWS Per ulteriori informazioni sugli endpoint dual-stack, che supportano sia il traffico che il IPv6 traffico, consulta [Using IPv4 Using Amazon S3 dual-stack endpoint nella Amazon Simple Storage Service User Guide](#). Gli endpoint dual-stack sono disponibili per alcuni servizi in alcune regioni. Se non esiste un endpoint dual-stack per il servizio o se la richiesta ha esito negativo. Regione AWS Questa opzione è disabilitata per impostazione predefinita.

Questa impostazione e l'impostazione `use_accelerate_endpoint` si escludono a vicenda.

Le impostazioni di configurazione degli endpoint si trovano in più posizioni, ad esempio nelle variabili di sistema o di ambiente utente, nei file di AWS configurazione locali o sono dichiarate esplicitamente nella riga di comando come parametro. Le impostazioni di configurazione AWS CLI degli endpoint hanno la precedenza nell'ordine seguente:

1. L'opzione della `--endpoint-url` riga di comando.
2. Se abilitata, la variabile di ambiente `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` globale dell'endpoint o l'impostazione del profilo `ignore_configure_endpoint_urls` per ignorare gli endpoint personalizzati.
3. Il valore fornito da una variabile `AWS_ENDPOINT_URL_<SERVICE>` di ambiente specifica del servizio, ad esempio. `AWS_ENDPOINT_URL_DYNAMODB`
4. I valori forniti dalle variabili `AWS_USE_DUALSTACK_ENDPOINT`, `AWS_USE_FIPS_ENDPOINT`, e di `AWS_ENDPOINT_URL` ambiente.
5. Il valore dell'endpoint specifico del servizio fornito dall'`endpoint_url` impostazione all'interno di una `services` sezione del file condiviso. `config`
6. Il valore fornito dall'`endpoint_url` impostazione all'interno di uno `profile` dei file condivisi. `config`
7. `use_dualstack_endpoint`, `use_fips_endpoint` e `endpoint_url` impostazioni.
8. Qualsiasi endpoint predefinito URL per il rispettivo dispositivo Servizio AWS viene utilizzato per ultimo. Per un elenco degli endpoint del servizio standard disponibili in ogni regione, consulta [Regioni ed endpoint AWS](#) nella Riferimenti generali di Amazon Web Services.

### *use\_fips\_endpoint*

Alcuni AWS servizi offrono endpoint che supportano il [Federal Information Processing Standard \(FIPS\) 140-2](#), in altri. Regioni AWS Se il AWS servizio supporta FIPS, questa impostazione specifica l'FIPS endpoint da utilizzare. AWS CLI A differenza degli AWS endpoint standard, gli FIPS endpoint utilizzano una libreria TLS software conforme a 140-2. FIPS Questi endpoint potrebbero essere necessari ad aziende che interagiscono con il governo degli Stati Uniti.

Se questa impostazione è abilitata, ma non esiste un FIPS endpoint per il servizio in uso Regione AWS, il comando potrebbe avere esito negativo. AWS In questo caso, specifica manualmente l'endpoint da utilizzare nel comando utilizzando l'`--endpoint-url` opzione o utilizza endpoint specifici del [servizio](#).

[Per ulteriori informazioni sulla specificazione degli endpoint in base al servizio, FIPS consulta Endpoints by Service Regione AWS. FIPS](#)

Le impostazioni di configurazione degli endpoint si trovano in più posizioni, ad esempio le variabili di sistema o di ambiente utente, i file di AWS configurazione locali o sono dichiarate esplicitamente nella riga di comando come parametro. Le impostazioni di configurazione AWS CLI degli endpoint hanno la precedenza nell'ordine seguente:

1. L'opzione della `--endpoint-url` riga di comando.
2. Se abilitata, la variabile di ambiente `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` globale dell'endpoint o l'impostazione del profilo `ignore_configure_endpoint_urls` per ignorare gli endpoint personalizzati.
3. Il valore fornito da una variabile `AWS_ENDPOINT_URL_<SERVICE>` di ambiente specifica del servizio, ad esempio. `AWS_ENDPOINT_URL_DYNAMODB`
4. I valori forniti dalle variabili `AWS_USE_DUALSTACK_ENDPOINT`, `AWS_USE_FIPS_ENDPOINT`, e di `AWS_ENDPOINT_URL` ambiente.
5. Il valore dell'endpoint specifico del servizio fornito dall'`endpoint_url` impostazione all'interno di una `services` sezione del file condiviso. `config`
6. Il valore fornito dall'`endpoint_url` impostazione all'interno di uno `profile` dei file condivisi. `config`
7. `use_dualstack_endpoint`, `use_fips_endpoint` e `endpoint_url` impostazioni.
8. Qualsiasi endpoint predefinito URL per il rispettivo dispositivo Servizio AWS viene utilizzato per ultimo. Per un elenco degli endpoint del servizio standard disponibili in ogni regione, consulta [Regioni ed endpoint AWS](#) nella Riferimenti generali di Amazon Web Services.

### `web_identity_token_file`

Specifica il percorso di un file che contiene un token di accesso OAuth 2.0 o un token ID OpenID Connect fornito da un provider di identità. L' AWS CLI carica il contenuto di questo file e lo passa come argomento `WebIdentityToken` all'operazione `AssumeRoleWithWebIdentity`.

La variabile di ambiente ha la precedenza `AWS_WEB_IDENTITY_TOKEN_FILE` su questa impostazione.

Per ulteriori informazioni sull'utilizzo delle identità Web, vedere. [the section called "Assunzione di un ruolo con un'identità Web"](#)

### `tcp_keepalive`

Specifica se il AWS CLI client utilizza pacchetti TCP keep-alive.

Questa voce non dispone di una variabile di ambiente o di un'opzione della riga di comando equivalente.

```
tcp_keepalive = false
```



## Impostazioni dei comandi S3 personalizzati

Amazon S3 supporta diverse impostazioni che configurano il modo in cui AWS CLI esegue le operazioni di Amazon S3. Alcune si applicano a tutti i comandi S3 in entrambi gli spazi dei nomi `s3api` e `s3`. Altre sono specifiche per i comandi «personalizzati» di S3 che astraggono le operazioni comuni e fanno più di una one-to-one mappatura a un'operazione. API I comandi di trasferimento `aws s3 cp`, `sync`, `mv` e `rm` dispongono di impostazioni aggiuntive che puoi utilizzare per controllare i trasferimenti S3.

Tutte queste opzioni possono essere configurate specificando l'impostazione nidificata `s3` nel file `config`. A ogni impostazione viene quindi applicato il rientro nella riga.

### Note

Queste impostazioni sono completamente facoltative. Dovresti essere in grado di utilizzare i comandi di trasferimento `aws s3` senza configurare tali impostazioni, le quali vengono rese disponibili per consentirti di ottenere prestazioni ottimizzate o prendere in considerazione l'ambiente specifico in cui esegui questi comandi `aws s3`.

Queste impostazioni vengono tutte configurate in una chiave `s3` di livello superiore nel file `config`, come mostrato nell'esempio seguente per il profilo `development`:

```
[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
```

Le impostazioni seguenti si applicano a qualsiasi comando S3 negli spazi dei nomi `s3` o `s3api`.

### **addressing\_style**

Specifica quale stile di indirizzamento utilizzare. Questo controlla se il nome del bucket è nel nome host o fa parte di. URL I valori validi sono `path`, `virtual` e `auto`. Il valore predefinito è `auto`.

Esistono due stili di creazione di un endpoint Amazon S3. Il primo è denominato `virtual` e include il nome del bucket come parte del nome host. Ad esempio: `https://bucketname.s3.amazonaws.com`. In alternativa, con lo path stile, si tratta il nome del bucket come se fosse un percorso inURI; ad esempio, `https://s3.amazonaws.com/bucketname` Il valore predefinito in CLI è `useauto`, che tenta di utilizzare lo `virtual` stile dove possibile, ma ricorre allo path stile quando necessario. Ad esempio, se il nome del bucket non è DNS compatibile, il nome del bucket non può far parte del nome host e deve essere nel percorso. Conauto, CLI rileverà questa condizione e passerà automaticamente path allo stile per te. Se imposti lo stile di indirizzamento `supath`, devi quindi assicurarti che la AWS regione che hai configurato AWS CLI corrisponda alla regione del tuo bucket.

### **payload\_signing\_enabled**

Specifica se SHA256 firmare i payload sigv4. Per impostazione predefinita, questa opzione è disabilitata per i caricamenti in streaming (`UploadPart`) durante l'utilizzo. `PutObject` HTTPS Per impostazione predefinita, è impostato su `false` per i caricamenti in streaming (`UploadPartPutObject`), ma solo se `ContentMD5` è presente a (viene generato per impostazione predefinita) e l'endpoint lo utilizza. HTTPS

Se impostato su `true`, le richieste S3 ricevono un'ulteriore convalida del contenuto sotto forma di un SHA256 checksum calcolato automaticamente e incluso nella firma della richiesta. Se impostata su `false`, il checksum non viene calcolato. Può essere utile disabilitare questa impostazione per ridurre il sovraccarico in termini di prestazioni dovuto al calcolo del checksum.

### **use\_accelerate\_endpoint**

Usa l'endpoint Amazon S3 Accelerate per tutti i `s3` comandi. `s3api` Il valore predefinito è `false`. Questa impostazione e l'impostazione `use_dualstack_endpoint` si escludono a vicenda.

Se impostato su `true`, AWS CLI indirizza tutte le richieste Amazon S3 `S3 Accelerate` all'endpoint a `s3-accelerate.amazonaws.com` Per usare questo endpoint, dovrai abilitare l'utilizzo di `S3 Accelerate` per il bucket. Tutte le richieste vengono inviate utilizzando lo stile virtuale di indirizzamento al bucket: `my-bucket.s3-accelerate.amazonaws.com`. Le richieste `ListBuckets`, `CreateBucket` e `DeleteBucket` non verranno inviate all'endpoint di accelerazione S3 in quanto tale endpoint non supporta queste operazioni. Questo comportamento può essere configurato anche se il parametro `--endpoint-url` è impostato su `https://s3-accelerate.amazonaws.com` o `http://s3-accelerate.amazonaws.com` per qualsiasi comando `s3` o `s3api`.

Le impostazioni seguenti si applicano solo ai comandi nel set di comandi dello spazio dei nomi `s3`:

## **max\_bandwidth**

Specifica la larghezza di banda massima che può essere consumata per caricare e scaricare dati da e verso Amazon S3. Il valore predefinito è senza limiti.

Ciò limita la larghezza di banda massima che i comandi S3 possono utilizzare per trasferire dati da e verso Amazon S3. Tale valore viene applicato solo ai caricamenti e ai download, non alle operazioni di copia o eliminazione. Il valore viene espresso in byte al secondo. Il valore può essere specificato come segue:

- Come un intero, `1048576` imposta ad esempio l'utilizzo massimo della larghezza di banda su 1 MB al secondo.
- Come un intero seguito da un suffisso di velocità, che può essere specificato utilizzando `KB/s`, `MB/s` o `GB/s`. Ad esempio, `300KB/s`, `10MB/s`.

In generale, ti consigliamo di provare in primo luogo a ridurre il consumo di larghezza di banda riducendo `max_concurrent_requests`. Se ciò non è sufficiente per limitare il consumo di larghezza di banda in modo appropriato alle tue esigenze, puoi utilizzare l'impostazione `max_bandwidth` per limitarlo ulteriormente. Ciò accade perché `max_concurrent_requests` controlla il numero di thread attualmente in esecuzione. Se invece prima si abbassa `max_bandwidth` ma si lascia un'impostazione `max_concurrent_requests` alta, ciò può comportare che i thread debbano attendere inutilmente. Ciò può portare a un consumo eccessivo di risorse e a timeout di connessione.

## **max\_concurrent\_requests**

Specifica il numero massimo di richieste simultanee. Il valore predefinito è 10.

I comandi di trasferimento `aws s3` sono a più thread. In qualsiasi momento, possono essere eseguite più richieste Amazon S3. Ad esempio, quando usi il comando `aws s3 cp localdir s3://bucket/ --recursive` per caricare file in un bucket S3, AWS CLI possono caricare i file `localdir/file1` e in `localdir/file3` parallelamente. `localdir/file2` L'impostazione `max_concurrent_requests` specifica il numero massimo di operazioni di trasferimento che possono essere eseguite nello stesso momento.

Potrebbe essere necessario modificare questo valore per vari motivi:

- Riduzione di questo valore: in alcuni ambienti, l'impostazione predefinita di 10 richieste simultanee può sovraccaricare il sistema. Questo può causare un timeout di connessione o

una riduzione della velocità di risposta del sistema. Se si riduce questo valore, i comandi di trasferimento S3 utilizzano un numero minore di risorse. Il completamento dei trasferimenti S3 può tuttavia richiedere più tempo. La riduzione di questo valore potrebbe essere necessaria se utilizzi uno strumento per limitare la larghezza di banda.

- Aumento di questo valore: in alcuni scenari, potresti desiderare che i trasferimenti di Amazon S3 vengano completati il più rapidamente possibile, utilizzando tutta la larghezza di banda di rete necessaria. In questo scenario, il numero predefinito di richieste simultanee potrebbe non essere sufficiente per utilizzare tutta la larghezza di banda di rete disponibile. L'aumento di questo valore può ridurre il tempo necessario per completare un trasferimento Amazon S3.

## **max\_queue\_size**

Specifica il numero massimo di attività nella coda. Il valore predefinito è 1000.

AWS CLI internamente utilizza un modello in cui mette in coda le attività di Amazon S3 che vengono poi eseguite da consumatori il cui numero è limitato a `max_concurrent_requests`. Un'attività è generalmente associata a una singola operazione di Amazon S3. Un'attività potrebbe ad esempio essere `PutObjectTask`, `GetObjectTask` o `UploadPartTask`. La velocità con cui le attività vengono aggiunte alla coda può essere molto maggiore rispetto a quella con cui i consumatori terminano le attività. Per evitare la crescita illimitata, esiste un limite specifico per la dimensione della coda delle attività. Questa impostazione consente di modificare il valore del numero massimo.

In generale, non è necessario modificare questa impostazione. Questa impostazione corrisponde anche al numero di attività di cui AWS CLI è a conoscenza e che devono essere eseguite. Ciò significa che per impostazione predefinita AWS CLI possono visualizzare solo 1000 attività future. L'aumento di questo valore significa che AWS CLI può conoscere più rapidamente il numero totale di attività necessarie, supponendo che la velocità di attesa sia superiore alla velocità di completamento delle attività. Il compromesso è che un valore `max_queue_size` maggiore richiede più memoria.

## **multipart\_chunksize**

Specifica la dimensione del blocco AWS CLI utilizzata per i trasferimenti multipart di singoli file. Il valore predefinito è 8 MB, con un minimo di 5 MB.

Quando un trasferimento di file supera `multipart_threshold`, AWS CLI divide il file in blocchi di questa dimensione. Questo valore può essere specificato utilizzando la stessa sintassi di `multipart_threshold`, come numero di byte sotto forma di intero o utilizzando una dimensione e un suffisso.

## `multipart_threshold`

Specifica la soglia di dimensione AWS CLI utilizzata per i trasferimenti multipart di singoli file. Il valore predefinito è 8 MB.

Quando si carica, si scarica o si copia un file, i comandi di Amazon S3 passano a operazioni in più parti se il file supera tale dimensione. Puoi specificare questo valore in uno dei due modi seguenti:

- Dimensione dei file in byte. Ad esempio 1048576.
- Dimensione dei file con un suffisso per la dimensione. Puoi utilizzare KB, MB, GB o TB. Ad esempio, 10MB, 1GB.

### Note

S3 può imporre vincoli per i valori validi che possono essere utilizzati per operazioni in più parti. Per ulteriori informazioni, consulta la [documentazione di S3 Multipart Upload nella Guida](#) per l'utente di Amazon Simple Storage Service.

## Configurazione delle variabili di ambiente per AWS CLI

Le variabili di ambiente forniscono un altro modo per specificare le opzioni di configurazione e le credenziali e possono essere utili per la creazione di script.

### Precedenza delle opzioni

- Se specifichi un'opzione utilizzando una delle variabili di ambiente descritte in questo argomento, questa sostituisce qualsiasi valore caricato da un profilo nel file di configurazione.
- Se si specifica un'opzione utilizzando un parametro sulla AWS CLI riga di comando, questo sostituisce qualsiasi valore della variabile di ambiente corrispondente o di un profilo nel file di configurazione.

Per ulteriori informazioni sulla precedenza e su come AWS CLI determina le credenziali da utilizzare, vedere. [Configurazione delle impostazioni per AWS CLI](#)

### Argomenti

- [Come impostare le variabili di ambiente](#)
- [AWS CLI variabili di ambiente supportate](#)

## Come impostare le variabili di ambiente

L'esempio seguente mostra come configurare le variabili di ambiente per l'utente predefinito.

Linux or macOS

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

L'impostazione della variabile di ambiente modifica il valore utilizzato fino al termine della sessione della shell o finché non imposti la variabile su un valore diverso. Puoi rendere le variabili persistenti per le sessioni future impostandole nello script di avvio della shell.

Windows Command Prompt

Da impostare per tutte le sessioni

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

Se si utilizza [setx](#) per impostare una variabile di ambiente, il valore utilizzato viene modificato nella sessione del prompt dei comandi corrente e in tutte le sessioni del prompt dei comandi create dopo l'esecuzione del comando. Ciò non ha alcun impatto su altre shell di comando già in esecuzione quando esegui il comando. Potrebbe essere necessario riavviare il terminale per caricare le impostazioni.

Da impostare solo per la sessione corrente

Se si utilizza [set](#) per impostare una variabile di ambiente, il valore utilizzato viene modificato fino al termine della sessione del prompt dei comandi corrente o finché non imposti la variabile su un valore diverso.

```
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="wJa1rXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"  
PS C:\> $Env:AWS_DEFAULT_REGION="us-west-2"
```

Se impostate una variabile di ambiente al PowerShell prompt, come mostrato negli esempi precedenti, il valore viene salvato solo per la durata della sessione corrente. Per rendere persistente l'impostazione della variabile di ambiente in tutte PowerShell le sessioni del prompt dei comandi, memorizzatela utilizzando l'applicazione System nel Pannello di controllo. In alternativa, puoi impostare la variabile per tutte le PowerShell sessioni future aggiungendola al tuo PowerShell profilo. Consulta la [PowerShell documentazione](#) per ulteriori informazioni sulla memorizzazione delle variabili di ambiente o sulla loro persistenza tra le sessioni.

## AWS CLI variabili di ambiente supportate

AWS CLI supporta le seguenti variabili di ambiente.

### **AWS\_ACCESS\_KEY\_ID**

Specifica una chiave di AWS accesso associata a un IAM account.

Se definita, questa variabile di ambiente sostituisce il valore per l'impostazione del profilo `aws_access_key_id`. Non è possibile specificare l'ID chiave di accesso utilizzando un'opzione della riga di comando.

### **AWS\_CA\_BUNDLE**

Specifica il percorso di un pacchetto di certificati da utilizzare per HTTPS la convalida dei certificati.

Se definita, questa variabile di ambiente sostituisce il valore per l'impostazione del profilo [ca\\_bundle](#). Puoi sovrascrivere questa variabile di ambiente utilizzando il parametro della riga di comando `--ca-bundle`.

### **AWS\_CLI\_S3\_MV\_VALIDATE\_SAME\_S3\_PATHS**

Se i bucket di origine e di destinazione sono gli stessi quando si utilizza il `s3 mv` comando `custom`, il file o l'oggetto di origine può essere spostato su se stesso, con conseguente eliminazione accidentale del file o dell'oggetto di origine. La variabile e l'opzione `--validate-same-s3-paths` di `AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS` ambiente specificano se convalidare il punto di accesso ARNs o gli alias del punto di accesso nell'origine o nella destinazione di Amazon S3. URIs

**Note**

La convalida del percorso richiede chiamate aggiuntive. `s3 mv API`

**AWS\_CONFIG\_FILE**

Specifica la posizione del file AWS CLI utilizzato per archiviare i profili di configurazione. Il percorso predefinito è `~/.aws/config`.

Non è possibile specificare questo valore in un'impostazione profilo denominato o utilizzando un parametro della riga di comando.

**AWS\_DATA\_PATH**

Un elenco di directory aggiuntive da controllare al di fuori del percorso di ricerca integrato durante il caricamento AWS CLI dei `~/.aws/models` dati. L'impostazione di questa variabile di ambiente indica le directory aggiuntive da controllare prima di tornare al percorso di ricerca integrato. Le voci multiple devono essere separate con il `os.pathsep` carattere, che si trova `:` su Linux o macOS e `;` su Windows.

**AWS\_DEFAULT\_OUTPUT**

Specifica il [formato di output](#) da utilizzare.

Se definita, questa variabile di ambiente sostituisce il valore per l'impostazione del profilo `output`. Puoi sovrascrivere questa variabile di ambiente utilizzando il parametro della riga di comando `--output`.

**AWS\_DEFAULT\_REGION**

Per impostazione predefinita, `Default region name` identifica la AWS regione ai cui server desideri inviare le richieste. Questa è tipicamente la regione più vicina per l'utente, ma può essere qualsiasi regione. Ad esempio, puoi digitare `us-west-2` per utilizzare US West (Oregon). Questa è la regione a cui verranno inviate tutte le richieste successive, se non specificato altrimenti in un singolo comando.

**Note**

È necessario specificare una AWS regione quando si utilizza la AWS CLI, in modo esplicito o impostando una regione predefinita. Per un elenco delle regioni disponibili,



consulta [Regioni ed endpoint](#). I designatori di regione utilizzati da AWS CLI sono gli stessi nomi che vedi negli AWS Management Console URLs endpoint del servizio.

Se definita, questa variabile di ambiente sostituisce il valore per l'impostazione del profilo `region`. È possibile sovrascrivere questa variabile di ambiente utilizzando il parametro della `--region` riga di comando .

## **AWS\_EC2\_METADATA\_DISABLED**

Disattiva l'uso del servizio di metadati delle EC2 istanze Amazon (IMDS).

Se impostato su `true`, le credenziali o la configurazione dell'utente (come la regione) non vengono richieste. IMDS

## **AWS\_ENDPOINT\_URL**

Specifica l'endpoint utilizzato per tutte le richieste di servizio.

Le impostazioni di configurazione degli endpoint si trovano in più posizioni, ad esempio le variabili di ambiente del sistema o dell'utente, i file di AWS configurazione locali, oppure sono dichiarate esplicitamente nella riga di comando come parametro. Le impostazioni di configurazione AWS CLI degli endpoint hanno la precedenza nell'ordine seguente:

1. L'opzione della `--endpoint-url` riga di comando.
2. Se abilitata, la variabile di ambiente `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` globale dell'endpoint o l'impostazione del profilo `ignore_configure_endpoint_urls` per ignorare gli endpoint personalizzati.
3. Il valore fornito da una variabile `AWS_ENDPOINT_URL_<SERVICE>` di ambiente specifica del servizio, ad esempio. `AWS_ENDPOINT_URL_DYNAMODB`
4. I valori forniti dalle variabili `AWS_USE_DUALSTACK_ENDPOINT`, `AWS_USE_FIPS_ENDPOINT`, e di `AWS_ENDPOINT_URL` ambiente.
5. Il valore dell'endpoint specifico del servizio fornito dall'`endpoint_url` impostazione all'interno di una `services` sezione del file condiviso. `config`
6. Il valore fornito dall'`endpoint_url` impostazione all'interno di uno `profile` dei file condivisi. `config`
7. `use_dualstack_endpoint`, `use_fips_endpoint` e `endpoint_url` impostazioni.

8. Qualsiasi endpoint predefinito URL per il rispettivo dispositivo Servizio AWS viene utilizzato per ultimo. Per un elenco degli endpoint del servizio standard disponibili in ogni regione, consulta [Regioni ed endpoint AWS](#) nella Riferimenti generali di Amazon Web Services.

## **AWS\_ENDPOINT\_URL\_<SERVICE>**

Specifica un endpoint personalizzato utilizzato per un servizio specifico, dove <SERVICE> viene sostituito con l'identificatore. Servizio AWS Ad esempio, Amazon DynamoDB ha un valore di `serviceId` [DynamoDB](#) Per questo servizio, la variabile di URL ambiente dell'endpoint è `AWS_ENDPOINT_URL_DYNAMODB`.

Per un elenco di tutte le variabili di ambiente specifiche del servizio, vedere. [Elenco di identificatori specifici del servizio](#)

Le impostazioni di configurazione degli endpoint si trovano in più posizioni, ad esempio le variabili di sistema o di ambiente utente, i file di AWS configurazione locali, oppure sono dichiarate esplicitamente nella riga di comando come parametro. Le impostazioni di configurazione AWS CLI degli endpoint hanno la precedenza nell'ordine seguente:

1. L'opzione della [--endpoint-url](#) riga di comando.
2. Se abilitata, la variabile di ambiente [AWS\\_IGNORE\\_CONFIGURED\\_ENDPOINT\\_URLS](#) globale dell'endpoint o l'impostazione del profilo [ignore\\_configure\\_endpoint\\_urls](#) per ignorare gli endpoint personalizzati.
3. Il valore fornito da una variabile [AWS\\_ENDPOINT\\_URL\\_<SERVICE>](#) di ambiente specifica del servizio, ad esempio. `AWS_ENDPOINT_URL_DYNAMODB`
4. I valori forniti dalle variabili [AWS\\_USE\\_DUALSTACK\\_ENDPOINT](#), [AWS\\_USE\\_FIPS\\_ENDPOINT](#), e di [AWS\\_ENDPOINT\\_URL](#) ambiente.
5. Il valore dell'endpoint specifico del servizio fornito dall'[endpoint\\_url](#) impostazione all'interno di una `services` sezione del file condiviso. `config`
6. Il valore fornito dall'[endpoint\\_url](#) impostazione all'interno di uno `profile` dei file condivisi. `config`
7. [use\\_dualstack\\_endpoint](#), [use\\_fips\\_endpoint](#) e [endpoint\\_url](#) impostazioni.
8. Qualsiasi endpoint predefinito URL per il rispettivo dispositivo Servizio AWS viene utilizzato per ultimo. Per un elenco degli endpoint del servizio standard disponibili in ogni regione, consulta [Regioni ed endpoint AWS](#) nella Riferimenti generali di Amazon Web Services.

## **AWS\_IGNORE\_CONFIGURED\_ENDPOINT\_URLS**

Se abilitato, AWS CLI ignora tutte le configurazioni personalizzate degli endpoint. I valori validi sono **true** e **false**.

Le impostazioni di configurazione degli endpoint si trovano in più posizioni, ad esempio nelle variabili di sistema o di ambiente utente, nei file di AWS configurazione locali, oppure sono dichiarate esplicitamente nella riga di comando come parametro. Le impostazioni di configurazione AWS CLI degli endpoint hanno la precedenza nell'ordine seguente:

1. L'opzione della `--endpoint-url` riga di comando.
2. Se abilitata, la variabile di ambiente `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` globale dell'endpoint o l'impostazione del profilo `ignore_configure_endpoint_urls` per ignorare gli endpoint personalizzati.
3. Il valore fornito da una variabile `AWS_ENDPOINT_URL_<SERVICE>` di ambiente specifica del servizio, ad esempio. `AWS_ENDPOINT_URL_DYNAMODB`
4. I valori forniti dalle variabili `AWS_USE_DUALSTACK_ENDPOINT`, `AWS_USE_FIPS_ENDPOINT`, e di `AWS_ENDPOINT_URL` ambiente.
5. Il valore dell'endpoint specifico del servizio fornito dall'`endpoint_url` impostazione all'interno di una `services` sezione del file condiviso. `config`
6. Il valore fornito dall'`endpoint_url` impostazione all'interno di uno `profile` dei file condivisi. `config`
7. `use_dualstack_endpoint`, `use_fips_endpoint` e `endpoint_url` impostazioni.
8. Qualsiasi endpoint predefinito URL per il rispettivo dispositivo Servizio AWS viene utilizzato per ultimo. Per un elenco degli endpoint del servizio standard disponibili in ogni regione, consulta [Regioni ed endpoint AWS](#) nella Riferimenti generali di Amazon Web Services.

## **AWS\_MAX\_ATTEMPTS**

Specifica il valore massimo di tentativi utilizzato dal gestore dei tentativi, in AWS CLI base al quale la chiamata iniziale viene conteggiata ai fini del valore fornito. Per ulteriori informazioni sui tentativi, consultare [AWS CLI nuovi tentativi nel AWS CLI](#).

Se definita, questa variabile di ambiente sostituisce il valore per l'impostazione dei profili.  
`max_attempts`

## **AWS\_METADATA\_SERVICE\_NUM\_ATTEMPTS**

Quando tenta di recuperare le credenziali su un'EC2istanza Amazon configurata con un IAM ruolo, AWS CLI tenta di recuperare le credenziali una volta dal servizio di metadati dell'istanza

prima di interrompersi. Se sai che i tuoi comandi verranno eseguiti su un'EC2istanza Amazon, puoi aumentare questo valore per AWS CLI riprovare più volte prima di rinunciare.

## **AWS\_METADATA\_SERVICE\_TIMEOUT**

Il numero di secondi prima che una connessione al servizio di metadati dell'istanza debba scadere. Per impostazione predefinita, quando si tenta di recuperare le credenziali su un'EC2istanza Amazon configurata con un IAM ruolo, la connessione al servizio di metadati dell'istanza scade dopo 1 secondo. Se sai di essere in esecuzione su un'EC2istanza Amazon con un IAM ruolo configurato, puoi aumentare questo valore se necessario.

## **AWS\_PROFILE**

Specificate il nome del AWS CLI profilo con le credenziali e le opzioni da utilizzare. Può essere il nome di un profilo archiviato in un file `credentials` o `config`, oppure il valore `default` per utilizzare il profilo di default.

Se definita, questa variabile di ambiente sostituisce il comportamento di utilizzo del profilo denominato `[default]` nel file di configurazione. Puoi sovrascrivere questa variabile di ambiente utilizzando il parametro della riga di comando `--profile`.

## **AWS\_RETRY\_MODE**

Specifica la modalità di riprova utilizzata. AWS CLI Sono disponibili tre modalità di riprova: `legacy` (predefinita), `standard` e `adattiva`. Per ulteriori informazioni sui tentativi, consultare [AWS CLI nuovi tentativi nel AWS CLI](#).

Se definita, questa variabile di ambiente sostituisce il valore per l'impostazione dei profili.  
`retry_mode`

## **AWS\_ROLE\_ARN**

Specifica l'Amazon Resource Name (ARN) di un IAM ruolo con un provider di identità web che desideri utilizzare per eseguire i AWS CLI comandi.

Utilizzato con le variabili di `AWS_ROLE_SESSION_NAME` ambiente `AWS_WEB_IDENTITY_TOKEN_FILE` e.

Se definita, questa variabile di ambiente sostituisce il valore per l'impostazione del profilo [role\\_arn](#). Non è possibile specificare un nome di sessione del ruolo come un parametro della riga di comando.

**Note**

Questa variabile di ambiente si applica solo a un ruolo assunto con Web Identity Provider e non si applica alla configurazione generale del provider Assume Role.

Per ulteriori informazioni sull'utilizzo delle identità Web, vedere [the section called “Assunzione di un ruolo con un'identità Web”](#).

**AWS\_ROLE\_SESSION\_NAME**

Specifica il nome da associare alla sessione del ruolo. Questo valore viene fornito al RoleSessionName parametro quando AWS CLI chiama l'AssumeRole operazione e diventa parte del ruolo assunto userARN: `arn:aws:sts::123456789012:assumed-role/role_name/role_session_name`. Si tratta di un parametro facoltativo. Se non fornisci questo valore, viene automaticamente generato un nome della sessione. Questo nome viene visualizzato nei AWS CloudTrail registri delle voci associate a questa sessione.

Se definita, questa variabile di ambiente sostituisce il valore per l'impostazione del profilo [role\\_session\\_name](#).

Utilizzato con le variabili di AWS\_WEB\_IDENTITY\_TOKEN\_FILE ambiente AWS\_ROLE\_ARN e.

Per ulteriori informazioni sull'utilizzo delle identità Web, vedere [the section called “Assunzione di un ruolo con un'identità Web”](#).

**Note**

Questa variabile di ambiente si applica solo a un ruolo assunto con Web Identity Provider e non si applica alla configurazione generale del provider Assume Role.

**AWS\_SDK\_UA\_APP\_ID**

Una singola Account AWS può essere utilizzata da più applicazioni clienti verso cui effettuare chiamate Servizi AWS. L'ID dell'applicazione identifica quale applicazione di origine ha effettuato una serie di chiamate utilizzando un Servizio AWS. AWS SDK e i servizi non utilizzano o interpretano questo valore se non per riemergere nelle comunicazioni con i clienti. Ad esempio, questo valore può essere incluso nelle e-mail operative per identificare in modo univoco quale delle applicazioni è associata alla notifica.

Per impostazione predefinita, non è presente alcun valore.

L'Application ID è una stringa con una lunghezza massima di 50 caratteri. Sono consentite lettere, numeri e i seguenti caratteri speciali:

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

Se definita, questa variabile di ambiente sostituisce il valore per l'impostazione del profilo [sdk\\_ua\\_app\\_id](#). Non è possibile specificare l'ID dell'applicazione come opzione della riga di comando.

## **AWS\_SECRET\_ACCESS\_KEY**

Specifica la chiave segreta associata alla chiave di accesso. Si tratta essenzialmente della "password" per la chiave di accesso.

Se definita, questa variabile di ambiente sostituisce il valore per l'impostazione del profilo `aws_secret_access_key`. Non è possibile specificare l'ID della chiave di accesso segreta come opzione della riga di comando.

## **AWS\_SESSION\_TOKEN**

Specifica il valore del token di sessione richiesto se si utilizzano credenziali di sicurezza temporanee recuperate direttamente dalle operazioni AWS STS . Per ulteriori informazioni, vedere la [sezione Output del comando assume-role nel Command Reference](#).AWS CLI

Se definita, questa variabile di ambiente sostituisce il valore per l'impostazione del profilo `aws_session_token`.

## **AWS\_SHARED\_CREDENTIALS\_FILE**

Specifica la posizione del file AWS CLI utilizzato per archiviare le chiavi di accesso. Il percorso predefinito è `~/.aws/credentials`.

Non è possibile specificare questo valore in un'impostazione profilo denominato o utilizzando un parametro della riga di comando.

## **AWS\_STS\_REGIONAL\_ENDPOINTS**

Specifica in che modo AWS CLI determina l'endpoint del AWS servizio utilizzato dal AWS CLI client per comunicare con (). AWS Security Token Service AWS STS Il valore predefinito per la AWS CLI versione 1 è `legacy`

È possibile specificare uno dei due valori:

- **legacy**— Utilizza l'endpoint globale `sts.amazonaws.com`, per le seguenti AWS regioni: `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`, `us-west-2`. Tutte le altre regioni utilizzano automaticamente il rispettivo endpoint regionale.
- **regional**— Utilizza AWS CLI sempre l'AWS STS endpoint per la regione attualmente configurata. Ad esempio, se il client è configurato per l'us-west-2, tutte le chiamate a AWS STS vengono effettuate all'endpoint regionale `sts.us-west-2.amazonaws.com` anziché all'endpoint globale `sts.amazonaws.com`. Per inviare una richiesta all'endpoint globale mentre questa impostazione è abilitata, è possibile impostare l'area geografica su `aws-global`.

## AWS\_USE\_DUALSTACK\_ENDPOINT

Abilita l'uso di endpoint dual-stack per inviare richieste. AWS Per ulteriori informazioni sugli endpoint dual-stack, che supportano sia il traffico che il IPv6 traffico, consulta Using IPv4 Using Amazon [S3 dual-stack endpoint nella Amazon Simple Storage Service User Guide](#). Gli endpoint dual-stack sono disponibili per alcuni servizi in alcune regioni. Se non esiste un endpoint dual-stack per il servizio o se la richiesta ha esito negativo. Regione AWS Questa opzione è disabilitata per impostazione predefinita.

Le impostazioni di configurazione degli endpoint si trovano in più posizioni, ad esempio nelle variabili di sistema o di ambiente utente, nei file di AWS configurazione locali o sono dichiarate esplicitamente nella riga di comando come parametro. Le impostazioni di configurazione AWS CLI degli endpoint hanno la precedenza nell'ordine seguente:

1. L'opzione della `--endpoint-url` riga di comando.
2. Se abilitata, la variabile di ambiente `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` globale dell'endpoint o l'impostazione del profilo `ignore_configure_endpoint_urls` per ignorare gli endpoint personalizzati.
3. Il valore fornito da una variabile `AWS_ENDPOINT_URL_<SERVICE>` di ambiente specifica del servizio, ad esempio. `AWS_ENDPOINT_URL_DYNAMODB`
4. I valori forniti dalle variabili `AWS_USE_DUALSTACK_ENDPOINT`, `AWS_USE_FIPS_ENDPOINT`, e di `AWS_ENDPOINT_URL` ambiente.
5. Il valore dell'endpoint specifico del servizio fornito dall'`endpoint_url` impostazione all'interno di una `services` sezione del file condiviso. `config`
6. Il valore fornito dall'`endpoint_url` impostazione all'interno di uno `profile` dei file condivisi. `config`

7. [use\\_dualstack\\_endpoint](#), [use\\_fips\\_endpoint](#), [endpoint\\_url](#) impostazioni.
8. Qualsiasi endpoint predefinito URL per il rispettivo dispositivo Servizio AWS viene utilizzato per ultimo. Per un elenco degli endpoint del servizio standard disponibili in ogni regione, consulta [Regioni ed endpoint AWS](#) nella Riferimenti generali di Amazon Web Services.

## AWS\_USE\_FIPS\_ENDPOINT

Alcuni AWS servizi offrono endpoint che supportano il [Federal Information Processing Standard \(FIPS\) 140-2](#), in altri. Regioni AWS Se il AWS servizio supporta FIPS, questa impostazione specifica l'FIPS endpoint da utilizzare. AWS CLI A differenza degli AWS endpoint standard, gli FIPS endpoint utilizzano una libreria TLS software conforme a 140-2. FIPS Questi endpoint potrebbero essere necessari ad aziende che interagiscono con il governo degli Stati Uniti.

Se questa impostazione è abilitata, ma non esiste un FIPS endpoint per il servizio in uso Regione AWS, il comando potrebbe avere esito negativo. AWS In questo caso, specifica manualmente l'endpoint da utilizzare nel comando utilizzando l'[--endpoint-url](#) opzione o utilizza endpoint specifici del [servizio](#).

[Per ulteriori informazioni sulla specificazione degli endpoint per, consulta FIPS Endpoints by Service Regione AWS. FIPS](#)

Le impostazioni di configurazione degli endpoint si trovano in più posizioni, ad esempio le variabili di sistema o di ambiente utente, i file di AWS configurazione locali o sono dichiarate esplicitamente nella riga di comando come parametro. Le impostazioni di configurazione AWS CLI degli endpoint hanno la precedenza nell'ordine seguente:

1. L'opzione della [--endpoint-url](#) riga di comando.
2. Se abilitata, la variabile di ambiente [AWS\\_IGNORE\\_CONFIGURED\\_ENDPOINT\\_URLS](#) globale dell'endpoint o l'impostazione del profilo [ignore\\_configure\\_endpoint\\_urls](#) per ignorare gli endpoint personalizzati.
3. Il valore fornito da una variabile [AWS\\_ENDPOINT\\_URL\\_<SERVICE>](#) di ambiente specifica del servizio, ad esempio. [AWS\\_ENDPOINT\\_URL\\_DYNAMODB](#)
4. I valori forniti dalle variabili [AWS\\_USE\\_DUALSTACK\\_ENDPOINT](#), [AWS\\_USE\\_FIPS\\_ENDPOINT](#), e di [AWS\\_ENDPOINT\\_URL](#) ambiente.
5. Il valore dell'endpoint specifico del servizio fornito dall'[endpoint\\_url](#) impostazione all'interno di una `services` sezione del file condiviso. `config`
6. Il valore fornito dall'[endpoint\\_url](#) impostazione all'interno di uno `profile` dei file condivisi. `config`



7. [use\\_dualstack\\_endpoint](#), [use\\_fips\\_endpoint](#) e [endpoint\\_url](#) impostazioni.
8. Qualsiasi endpoint predefinito URL per il rispettivo dispositivo Servizio AWS viene utilizzato per ultimo. Per un elenco degli endpoint del servizio standard disponibili in ogni regione, consulta [Regioni ed endpoint AWS](#) nella Riferimenti generali di Amazon Web Services.

## [AWS\\_WEB\\_IDENTITY\\_TOKEN\\_FILE](#)

Specifica il percorso di un file che contiene un token di accesso OAuth 2.0 o un token ID OpenID Connect fornito da un provider di identità. L' AWS CLI carica il contenuto di questo file e lo passa come argomento `WebIdentityToken` all'operazione `AssumeRoleWithWebIdentity`.

Utilizzato con le variabili di ambiente `AWS_ROLE_SESSION_NAME` e `AWS_ROLE_ARN`.

Se definita, questa variabile di ambiente sostituisce il valore per l'impostazione del profilo `web_identity_token_file`.

Per ulteriori informazioni sull'utilizzo delle identità Web, vedere [the section called "Assunzione di un ruolo con un'identità Web"](#).

### Note

Questa variabile di ambiente si applica solo a un ruolo assunto con Web Identity Provider e non si applica alla configurazione generale del provider Assume Role.

## Opzioni della riga di comando in AWS CLI

In AWS CLI, le opzioni della riga di comando sono parametri globali che è possibile utilizzare per sovrascrivere le impostazioni di configurazione predefinite, qualsiasi impostazione del profilo corrispondente o l'impostazione della variabile di ambiente per quel singolo comando. Non puoi utilizzare le opzioni della riga di comando per specificare direttamente le credenziali, anche se puoi specificare il profilo da utilizzare.

### Argomenti

- [Come utilizzare le opzioni della riga di comando](#)
- [AWS CLI opzioni della riga di comando globali supportate](#)
- [Usi comuni delle opzioni della riga di comando](#)

## Come utilizzare le opzioni della riga di comando

La maggior parte delle opzioni della riga di comando sono stringhe semplici, come il nome del profilo `profile1` nell'esempio seguente:

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

Ogni opzione che accetta un argomento richiede uno spazio o un simbolo di uguale ("=") che separa l'argomento dal nome dell'opzione. Se il valore dell'argomento è una stringa che contiene uno spazio, occorre racchiudere l'argomento tra virgolette. Per informazioni dettagliate sui tipi di argomenti e sulla formattazione dei parametri, vedere. [Specificazione dei valori dei parametri nel AWS CLI](#)

## AWS CLI opzioni della riga di comando globali supportate

In AWS CLI è possibile utilizzare le seguenti opzioni della riga di comando per sovrascrivere le impostazioni di configurazione predefinite, qualsiasi impostazione del profilo corrispondente o l'impostazione della variabile di ambiente per quel singolo comando.

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

Specifica il pacchetto di certificati dell'autorità di certificazione (CA) da utilizzare per la verifica dei certificati. SSL

Se definita, questa opzione sostituisce il valore per l'impostazione del profilo e la variabile di ambiente `ca_bundle`. [AWS\\_CA\\_BUNDLE](#)

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

Specifica il tempo di connessione socket massimo in secondi. Se il valore è impostato su zero (0), la connessione socket attende per un periodo di tempo indefinito (si blocca) e non si verifica alcun timeout.

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

Specifica il tempo di lettura del socket massimo in secondi. Se il valore è impostato su zero (0), la lettura del socket attende per un periodo di tempo indefinito (si blocca) e non si verifica alcun timeout.

**--colore <string>**

Specifica il supporto per l'output di colori. I valori validi sono on, off e auto. Il valore predefinito è auto.

**--debug**

Un interruttore booleano che abilita la registrazione di debug. Per AWS CLI impostazione predefinita, fornisce informazioni dettagliate su eventuali successi o fallimenti relativi ai risultati dei comandi nell'output del comando. L'--debugopzione fornisce i log completi di Python. Ciò include informazioni stderr diagnostiche aggiuntive sul funzionamento del comando che possono essere utili per la risoluzione dei motivi per cui un comando fornisce risultati imprevisti. Per visualizzare facilmente i log di debug, si consiglia di inviare i log in un file per facilitare la ricerca delle informazioni. È possibile eseguire questa operazione utilizzando uno dei seguenti metodi.

Per inviare solo le informazioni di stderr diagnostica, aggiungi 2> debug.txt debug.txt dov'è il nome che vuoi usare per il tuo file di debug:

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

Per inviare sia le informazioni di output che quelle di stderr diagnostica, aggiungi &> debug.txt dove debug.txt è il nome che vuoi usare per il tuo file di debug:

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

**--endpoint-url <string>**

Specifica a chi inviare la URL richiesta. Per la maggior parte dei comandi, determina AWS CLI automaticamente il URL valore in base al servizio selezionato e alla AWS regione specificata. Tuttavia, alcuni comandi richiedono la specificazione di un account specifico. URL Puoi anche configurare alcuni AWS servizi per [ospitare un endpoint direttamente all'interno del tuo account privato VPC](#), cosa che potrebbe quindi dover essere specificata.

Il seguente esempio di comando utilizza un endpoint Amazon S3 personalizzato. URL

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

Le impostazioni di configurazione degli endpoint si trovano in più posizioni, ad esempio nelle variabili di sistema o di ambiente utente, nei file di AWS configurazione locali, oppure

sono dichiarate esplicitamente nella riga di comando come parametro. Le impostazioni di configurazione AWS CLI degli endpoint hanno la precedenza nell'ordine seguente:

1. L'opzione della `--endpoint-url` riga di comando.
2. Se abilitata, la variabile di ambiente `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` globale dell'endpoint o l'impostazione del profilo `ignore_configure_endpoint_urls` per ignorare gli endpoint personalizzati.
3. Il valore fornito da una variabile `AWS_ENDPOINT_URL_<SERVICE>` di ambiente specifica del servizio, ad esempio. `AWS_ENDPOINT_URL_DYNAMODB`
4. I valori forniti dalle variabili `AWS_USE_DUALSTACK_ENDPOINT`, `AWS_USE_FIPS_ENDPOINT`, e di `AWS_ENDPOINT_URL` ambiente.
5. Il valore dell'endpoint specifico del servizio fornito dall'`endpoint_url` impostazione all'interno di una `services` sezione del file condiviso. `config`
6. Il valore fornito dall'`endpoint_url` impostazione all'interno di uno `profile` dei file condivisi. `config`
7. `use_dualstack_endpoint`, `use_fips_endpoint` e `endpoint_url` impostazioni.
8. Qualsiasi endpoint predefinito URL per il rispettivo dispositivo Servizio AWS viene utilizzato per ultimo. Per un elenco degli endpoint del servizio standard disponibili in ogni regione, consulta [Regioni ed endpoint AWS](#) nella Riferimenti generali di Amazon Web Services.

#### `--no-paginate`

Un interruttore booleano che disabilita le chiamate multiple esegue automaticamente la ricezione di tutti i risultati dei comandi AWS CLI che creano l'impaginazione dell'output. Ciò significa che viene visualizzata solo la prima pagina dell'output.

#### `--no-sign-request`

Un interruttore booleano che disabilita la firma delle HTTP richieste all'endpoint del AWS servizio. In questo modo si evita il caricamento delle credenziali.

#### `--no-verify-ssl`

Per impostazione predefinita, vengono AWS CLI utilizzati SSL quando si comunica con i servizi. AWS Per ogni SSL connessione e chiamata, AWS CLI verifica i SSL certificati. L'utilizzo di questa opzione sostituisce il comportamento predefinito di verifica dei certificati. SSL

**⚠ Warning**

Questa opzione non è una procedura consigliata. Se la utilizzi `--no-verify-ssl`, il traffico tra il cliente e AWS i servizi non è più sicuro. Ciò significa che il traffico rappresenta un rischio per la sicurezza ed è vulnerabile agli man-in-the-middle exploit. Se riscontri problemi con i certificati, è meglio risolvere invece tali problemi. Per la procedura di risoluzione dei problemi relativi ai certificati, consulta [the section called “SSL errori nei certificati”](#).

**--output *<string>***

Specifica il formato di output da utilizzare per questo comando. Puoi specificare uno qualsiasi dei valori seguenti:

- **[json](#)**— L'output è formattato come [JSON](#)stringa.
- **[text](#)**— L'output è formattato come più righe di valori di stringa separati da tabulazioni. Questo può essere utile per passare l'output a un elaboratore di testi, ad esempio `grep`, `sed` o `awk`.
- **[table](#)**— L'output viene formattato come tabella utilizzando i caratteri `+|-` per formare i bordi delle celle. In genere presenta le informazioni in un formato comprensibile molto più semplice da leggere rispetto ad altri, ma non altrettanto utile a livello programmatico.

**--profile *<string>***

Specifica il [profilo denominato](#) da utilizzare per questo comando. Per impostare profili denominati aggiuntivi, puoi utilizzare il comando `aws configure` con l'opzione `--profile`.

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

**--query *<string>***

Specifica una [JMESPathquery](#) da utilizzare per filtrare i dati di risposta. Per ulteriori informazioni, consulta [Uscita di filtraggio nel AWS CLI](#).

**--region *<string>***

Specifica a quale AWS regione inviare la richiesta di AWS questo comando. Per un elenco di tutte le regioni che puoi specificare, consulta [Regioni ed endpoint AWS](#) nella Riferimenti generali di Amazon Web Services.

## --versione

Un interruttore booleano che visualizza la versione corrente del AWS CLI programma in esecuzione.

## Usi comuni delle opzioni della riga di comando

Usi comuni per le opzioni della riga di comando includono il controllo delle risorse in più regioni AWS e la modifica del formato di output per maggiore leggibilità o facilità d'uso al momento della creazione di script. Negli esempi seguenti, eseguiamo il comando `describe-instances` su ciascuna regione finché non individuiamo in quale regione si trova la nostra istanza.

```
$ 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             |||
...

```

## Configurazione del completamento dei comandi in AWS CLI

Il AWS Command Line Interface (AWS CLI) include una funzionalità di completamento dei comandi compatibile con bash che consente di utilizzare il tasto Tab per completare un comando inserito parzialmente. Sulla maggior parte dei sistemi è necessario configurarla manualmente.

### Argomenti

- [Come funziona](#)

- [Configurazione del completamento dei comandi su Linux o macOS](#)
- [Configurazione del completamento dei comandi in Windows](#)

## Come funziona

Quando si immette parzialmente un comando, un parametro o un'opzione, la caratteristica di completamento dei comandi completa automaticamente il comando o visualizza un elenco suggerito di comandi. Per richiedere il completamento del comando, si immette parzialmente un comando e si preme il tasto di completamento, che in genere è *Tab* nella maggior parte delle conchiglie.

Negli esempi seguenti vengono illustrati diversi modi in cui è possibile utilizzare il completamento dei comandi:

- Inserisci parzialmente un comando e premi *Tab* per visualizzare un elenco di comandi suggerito.

```
$ 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
```

- Inserite parzialmente un parametro e premete *Tab* per visualizzare un elenco di parametri suggerito.

```
$ 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
```

- Immettete un parametro e premete *Tab* per visualizzare un elenco suggerito di valori delle risorse. Questa funzionalità è disponibile solo nella AWS CLI versione 2.

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

## Configurazione del completamento dei comandi su Linux o macOS

Per configurare il completamento dei comandi su Linux o macOS, devi conoscere il nome della shell che stai utilizzando e la posizione dello `aws_completer` script.

### Note

Il completamento dei comandi viene configurato e abilitato automaticamente per impostazione predefinita sulle EC2 istanze Amazon che eseguono Amazon Linux.

### Argomenti

- [Conferma che la cartella del completatore si trova nel tuo percorso](#)
- [Abilitazione del completamento dei comandi](#)
- [Verifica il completamento del comando](#)

### Conferma che la cartella del completatore si trova nel tuo percorso

AWS Affinché il completatore funzioni correttamente, `aws_completer` deve trovarsi nel percorso della shell. Il `which` comando può verificare se il completatore si trova nel percorso dell'utente.

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

Se il comando `which` non riesce a trovare il completatore, utilizza i seguenti passaggi per aggiungere la cartella del completatore al tuo percorso.

### Passaggio 1: individua il completatore AWS

La posizione del AWS completatore può variare a seconda del metodo di installazione utilizzato.

- Package Manager: programmi come `pip`, `yumbrew`, e `apt-get` in genere installano il AWS completatore (o un collegamento simbolico ad esso) in una posizione di percorso standard.
  - Se è stato utilizzato `pip` senza il `--user` parametro, il percorso predefinito è `/usr/local/bin/aws_completer`
  - Se è stato utilizzato `pip` con il `--user` parametro, il percorso predefinito è `home/username/.local/bin/aws_completer`.



- Programma di installazione in bundle: se hai utilizzato il programma di installazione in bundle, il percorso predefinito è `/usr/local/bin/aws_completer`

Se tutto il resto fallisce, puoi usare il `find` comando per cercare il programma di completamento nel file system. AWS

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

## Fase 2: Identifica la shell

Per identificare la shell in uso, è possibile utilizzare uno dei seguenti comandi.

- `echo $ SHELL` — Visualizza il nome del file di programma della shell. Generalmente corrisponde al nome della shell in uso, a meno che non venga avviata un'altra shell dopo avere effettuato l'accesso.

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

- `ps` — Visualizza i processi in esecuzione per l'utente corrente. Uno di questi è la shell.

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

## Passaggio 3: aggiungi il completatore al tuo percorso

1. Trova lo script del profilo della tua shell nella tua cartella utente.

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

- Bash — `.bash_profile`, o `.profile` `.bash_login`
- Cenero — `.zshrc`
- Tcsh — `.tcshrc` `.cshrc`, o `.login`

2. Aggiungi un comando di esportazione al termine dello script del profilo simile a quello dell'esempio seguente. Sostituisci `/usr/local/bin/` con la cartella individuata nella sezione precedente.

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

3. Ricarica il profilo nella sessione corrente per rendere effettive tali modifiche. Sostituisci `.bash_profile` con il nome dello script della shell individuato nella prima sezione.

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

## Abilitazione del completamento dei comandi

Dopo aver verificato che il completatore sia nel tuo percorso, abilita il completamento del comando eseguendo il comando appropriato per la shell che stai utilizzando. Puoi aggiungere il comando al profilo della shell per eseguirlo ogni volta che apri una nuova shell. In ogni comando, sostituisci il `/usr/local/bin/` percorso con quello trovato sul sistema in [Conferma che la cartella del completatore si trova nel tuo percorso](#).

- **bash**— Usa il comando integrato `complete`.

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

Aggiungi il comando precedente `~/.bashrc` a per eseguirlo ogni volta che apri una nuova shell. Per accertarti che il comando venga eseguito anche nelle shell di accesso, l'origine di `~/.bash_profile` deve essere `~/.bashrc`.

- **zsh**— Per eseguire il completamento del comando, è necessario eseguirlo `bashcompinit` aggiungendo la seguente riga di caricamento automatico alla fine dello script del `~/.zshrc` profilo.

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

Per abilitare il completamento dei comandi, utilizzare il comando integrato `complete`.

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

Aggiungi i comandi precedenti `~/.zshrc` a per eseguirlo ogni volta che apri una nuova shell.

- **tcsch**— Complete for tcsh richiede un tipo di parola e uno schema per definire il comportamento di completamento.

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

Aggiungi il comando precedente `~/ .tcschrc` a per eseguirlo ogni volta che apri una nuova shell.

Dopo aver abilitato il completamento del comando, [Verifica il completamento del comando](#) funziona.

## Verifica il completamento del comando

Dopo aver abilitato il completamento dei comandi, ricarica la shell, inserisci un comando parziale e premi Tab per vedere i comandi disponibili.

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

## Configurazione del completamento dei comandi in Windows

### Note

Per informazioni su come PowerShell gestisce il loro completamento, incluse le varie chiavi di completamento, vedi [about\\_Tab\\_Expansion in Microsoft Docs](#). PowerShell

Per abilitare il completamento dei PowerShell comandi per Windows, completa i passaggi seguenti in PowerShell

1. Apri il tuo `$PROFILE` con il seguente comando.

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

Se non ne hai uno `$PROFILE`, crea un profilo utente usando il seguente comando.

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

Per ulteriori informazioni sui PowerShell profili, vedi [Come usare i profili in Windows PowerShell ISE](#) sul sito Web Microsoft Docs.

2. Per abilitare il completamento dei comandi, aggiungi il seguente blocco di codice al tuo profilo, salva e chiudi il file.

```
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
}
```

3. Dopo aver abilitato il completamento dei comandi, ricarica la shell, inserisci un comando parziale e premi Tab per scorrere tra i comandi disponibili.

```
$ aws sTab
```

```
$ aws s3
```

Per vedere tutti i comandi disponibili disponibili al completamento, inserisci un comando parziale e premi Ctrl + Spazio.

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

## AWS CLI nuovi tentativi nel AWS CLI

In questo argomento viene descritto il modo in cui le chiamate ai AWS servizi AWS CLI potrebbero non riuscire a causa di problemi imprevisti. Questi problemi possono verificarsi sul lato server o

potrebbero non riuscire a causa della limitazione della velocità imposta dal AWS servizio che stai tentando di chiamare. Questi tipi di errori di solito non richiedono una gestione speciale e la chiamata viene riefettuata automaticamente, spesso dopo un breve periodo di attesa. AWS CLI Fornisce molte funzionalità che aiutano a ritentare le chiamate dei client ai AWS servizi quando si verificano questi tipi di errori o eccezioni.

## Argomenti

- [Modalità di riprova disponibili](#)
- [Configurazione di una modalità di ripetizione](#)
- [Visualizzazione dei registri dei tentativi di nuovo tentativo](#)

## Modalità di riprova disponibili

AWS CLI Ha diverse modalità tra cui scegliere a seconda della versione:

- [modalità Legacy Retry](#)
- [Modalità di riprova standard](#)
- [Modalità di ritentativo adattiva](#)

### modalità Legacy Retry

La modalità Legacy è la modalità predefinita utilizzata dalla AWS CLI versione 1. La modalità Legacy utilizza un vecchio gestore di tentativi con funzionalità limitate che includono:

- Un valore predefinito di 4 per il numero massimo di tentativi di ripetizione, per un totale di 5 tentativi di chiamata. Questo valore può essere sovrascritto tramite il parametro di `max_attempts` configurazione.
- DynamoDB ha un valore predefinito di 9 per il numero massimo di tentativi di ripetizione, per un totale di 10 tentativi di chiamata. Questo valore può essere sovrascritto tramite il parametro di configurazione. `max_attempts`
- Riprova i tentativi per il seguente numero limitato di errori/eccezioni:
  - `ConnectionError`
  - `ConnectionClosedError`
  - `ReadTimeoutError`

- `EndpointConnectionError`
- Errori ed eccezioni di limitazione/limitazione sul lato del servizio:
  - `Throttling`
  - `ThrottlingException`
  - `ThrottledException`
  - `RequestThrottledException`
  - `ProvisionedThroughputExceededException`
- Riprova con diversi codici di HTTP stato, tra cui 429, 500, 502, 503, 504 e 509.
- Qualsiasi nuovo tentativo includerà un backoff esponenziale di un fattore base pari a 2.

## Modalità di riprova standard

La modalità standard è un insieme standard di regole di nuovo tentativo AWS SDKs con più funzionalità rispetto alla versione precedente. La modalità standard è stata creata per la AWS CLI versione 2 ed è stata trasferita alla AWS CLI versione 1. Le funzionalità della modalità standard includono:

- Il valore predefinito è 2 per il numero massimo di tentativi di ripetizione, per un totale di 3 tentativi di chiamata. Questo valore può essere sovrascritto tramite il parametro di `max_attempts` configurazione.
- Riprova i tentativi per il seguente elenco esteso di errori/eccezioni:
  - Errori/eccezioni transitori
    - `RequestTimeout`
    - `RequestTimeoutException`
    - `PriorRequestNotComplete`
    - `ConnectionError`
    - `HTTPClientError`
  - Errori ed eccezioni di limitazione/limitazione sul lato del servizio:
    - `Throttling`
    - `ThrottlingException`
    - `ThrottledException`
    - `RequestThrottledException`

- `TooManyRequestsException`
  - `ProvisionedThroughputExceededException`
  - `TransactionInProgressException`
  - `RequestLimitExceeded`
  - `BandwidthLimitExceeded`
  - `LimitExceededException`
  - `RequestThrottled`
  - `SlowDown`
  - `EC2ThrottledException`
- Riprova con codici di errore temporanei e non descrittivi. In particolare, questi codici di HTTP stato: 500, 502, 503, 504.
  - Qualsiasi nuovo tentativo includerà un backoff esponenziale di un fattore base pari a 2 per un tempo di backoff massimo di 20 secondi.

## Modalità di ritentativo adattiva

### Warning

La modalità adattiva è una modalità sperimentale ed è soggetta a modifiche, sia nelle funzionalità che nel comportamento.

La modalità di riprova adattiva è una modalità di riprova sperimentale che include tutte le funzionalità della modalità standard. Oltre alle funzionalità della modalità standard, la modalità adattiva introduce anche la limitazione della velocità sul lato client mediante l'uso di un token bucket e di variabili rate-limit che vengono aggiornate dinamicamente a ogni nuovo tentativo. Questa modalità offre flessibilità nei tentativi lato client e si adatta alla risposta allo stato di errore/eccezione di un servizio. AWS

Ad ogni nuovo tentativo, la modalità adattiva modifica le variabili del limite di velocità in base all'errore, all'eccezione o al codice di stato presentato nella risposta del servizio. HTTP AWS Queste variabili relative al limite di velocità vengono quindi utilizzate per calcolare una nuova tariffa di chiamata per il client. Ogni HTTP risposta di eccezione/errore o insuccesso (fornita nell'elenco precedente) fornita da un AWS servizio aggiorna le variabili relative al limite di frequenza man mano

che si verificano nuovi tentativi fino al raggiungimento del successo, all'esaurimento del bucket di token o al raggiungimento del valore massimo di tentativi configurato.

## Configurazione di una modalità di ripetizione

AWS CLI Include una varietà di configurazioni di nuovi tentativi e metodi di configurazione da considerare durante la creazione dell'oggetto client.

### Metodi di configurazione disponibili

In AWS CLI, gli utenti possono configurare i nuovi tentativi nei seguenti modi:

- Variabili di ambiente
- AWS CLI file di configurazione

Gli utenti possono personalizzare le seguenti opzioni di riprova:

- Modalità Riprova: specifica la modalità di riprova utilizzata. AWS CLI Come descritto in precedenza, sono disponibili tre modalità di riprova: legacy, standard e adattiva. Il valore predefinito per la AWS CLI versione 1 è legacy. La standard.
- Numero massimo di tentativi: specifica il valore massimo di tentativi utilizzato dal gestore dei tentativi, in AWS CLI base al quale la chiamata iniziale viene conteggiata ai fini del valore fornito. Il valore predefinito è 5.

### Definizione di una configurazione di nuovo tentativo nelle variabili di ambiente

Per definire la configurazione dei nuovi tentativi per il AWS CLI, aggiorna le variabili di ambiente del sistema operativo.

Le variabili di ambiente Retry sono:

- AWS\_RETRY\_MODE
- AWS\_MAX\_ATTEMPTS

Per ulteriori informazioni sulle variabili di ambiente, vedere [Configurazione delle variabili di ambiente per AWS CLI](#).



## Visualizzazione dei registri dei tentativi di nuovo tentativo

AWS CLI Utilizza la metodologia di riprova e la registrazione di Boto3. È possibile utilizzare l'--debugopzione su qualsiasi comando per ricevere i registri di debug. Per ulteriori informazioni su come utilizzare l'--debugopzione, vedere. [Opzioni della riga di comando in AWS CLI](#)

Se cerchi «riprova» nei log di debug, troverai le informazioni sui nuovi tentativi di cui hai bisogno. Le voci del registro del client relative ai tentativi di nuovo tentativo dipendono dalla modalità di ritentativo abilitata.

### Modalità Legacy:

I messaggi di riprova vengono generati da botocore.retryhandler. Vedrai uno dei tre messaggi seguenti:

- No retry needed
- Retry needed, action of: *<action\_name>*
- Reached the maximum number of retry attempts: *<attempt\_number>*

### Modalità standard o adattiva:

I messaggi di riprova vengono generati da botocore.retries.standard. Vedrai uno dei tre messaggi seguenti:

- No retrying request
- Retry needed, retrying request after delay of: *<delay\_value>*
- Retry needed but retry quota reached, not retrying request

Per il file di definizione completo dei tentativi di botocore, consulta [\\_retry.json](#) nel repository botocore. GitHub

## Utilizzo di un HTTP proxy per AWS CLI

Per accedere AWS tramite server proxy, puoi configurare le HTTP\_PROXY variabili di HTTPS\_PROXY ambiente con i nomi di DNS dominio o gli indirizzi IP e i numeri di porta utilizzati dai server proxy.

### Argomenti

- [Utilizzo degli esempi](#)

- [Autenticazione in un proxy](#)
- [Utilizzo di un proxy su EC2 istanze Amazon](#)
- [Risoluzione dei problemi](#)

## Utilizzo degli esempi

### Note

Gli esempi seguenti mostrano il nome della variabile di ambiente in lettere maiuscole. Tuttavia, se si specifica una variabile due volte utilizzando lettere maiuscole e minuscole, le lettere minuscole hanno la precedenza. Si consiglia di definire ogni variabile una sola volta per evitare confusione nel sistema e comportamenti imprevisti.

Gli esempi seguenti mostrano come utilizzare l'indirizzo IP esplicito del proxy o un DNS nome che si risolva nell'indirizzo IP del proxy. Entrambi possono essere seguiti da due punti e dal numero di porta a cui devono essere inviate le query.

### 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

Da impostare per tutte le sessioni

```
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
```

Se si utilizza [setx](#) per impostare una variabile di ambiente, il valore utilizzato viene modificato nella sessione del prompt dei comandi corrente e in tutte le sessioni del prompt dei comandi create dopo l'esecuzione del comando. Ciò non ha alcun impatto su altre shell di comando già in esecuzione quando esegui il comando.

Da impostare solo per la sessione corrente

Se si utilizza [set](#) per impostare una variabile di ambiente, il valore utilizzato viene modificato fino al termine della sessione del prompt dei comandi corrente o finché non imposti la variabile su un valore diverso.

```
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
```

## Autenticazione in un proxy

### Note

AWS CLI Non supporta i NTLM proxy. [Se utilizzi un NTLM proxy del protocollo Kerberos, potresti riuscire a connetterti tramite un proxy di autenticazione come Cntlm.](#)

AWS CLI Supporta l'autenticazione di base. HTTP Specificare il nome utente e la password nel proxyURL, come segue.

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

Da impostare per tutte le sessioni

```
C:\> setx HTTP_PROXY http://username:password@proxy.example.com:1234
C:\> setx HTTPS_PROXY http://username:password@proxy.example.com:5678
```

Da impostare solo per la sessione corrente

```
C:\> set HTTP_PROXY=http://username:password@proxy.example.com:1234
C:\> set HTTPS_PROXY=http://username:password@proxy.example.com:5678
```

## Utilizzo di un proxy su EC2 istanze Amazon

Se configuri un proxy su un'EC2 istanza Amazon lanciata con un IAM ruolo associato, assicurati di esentare l'indirizzo utilizzato per accedere ai [metadati dell'istanza](#). Per fare ciò, imposta la variabile di NO\_PROXY ambiente sull'indirizzo IP del servizio di metadati dell'istanza, 169.254.169.254. Questo indirizzo non varia.

Linux or macOS

```
$ export NO_PROXY=169.254.169.254
```

Windows Command Prompt

Da impostare per tutte le sessioni

```
C:\> setx NO_PROXY 169.254.169.254
```

Da impostare solo per la sessione corrente

```
C:\> set NO_PROXY=169.254.169.254
```

## Risoluzione dei problemi

Se riscontri problemi con il AWS CLI, consulta la procedura [Risolvere gli errori](#) per la risoluzione dei problemi. Per le procedure di risoluzione dei problemi più importanti, consulta [the section called "SSL errori nei certificati"](#).

## Utilizzo degli endpoint in AWS CLI

Per connettersi a livello di codice a un Servizio AWS, si utilizza un endpoint. Un endpoint è il punto URL di ingresso per un servizio Web. AWS Il AWS Command Line Interface (AWS CLI) utilizza automaticamente l'endpoint predefinito per ogni servizio in un Regione AWS, ma è possibile specificare un endpoint alternativo per le richieste. API

Argomenti sugli endpoint

- [Imposta l'endpoint per un singolo comando](#)

- [Imposta un endpoint globale per tutti Servizi AWS](#)
- [Imposta per utilizzare gli endpoint per tutti FIPs Servizi AWS](#)
- [Imposta per utilizzare endpoint dual-stack per tutti Servizi AWS](#)
- [Imposta endpoint specifici del servizio](#)
  - [Endpoint specifici del servizio: variabili di ambiente](#)
  - [Endpoint specifici del servizio: file condiviso config](#)
  - [Endpoint specifici del servizio: elenco di identificatori specifici del servizio](#)
- [Priorità della configurazione e delle impostazioni degli endpoint](#)

## Imposta l'endpoint per un singolo comando

Per sovrascrivere le impostazioni degli endpoint o le variabili di ambiente per un singolo comando, utilizzate l'opzione della `--endpoint-url` riga di comando. Il seguente esempio di comando utilizza un endpoint Amazon S3 personalizzato. URL

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

## Imposta un endpoint globale per tutti Servizi AWS

Per indirizzare le richieste di tutti i servizi a un endpoint personalizzatoURL, utilizza una delle seguenti impostazioni:

- Variabili di ambiente:
  - [AWS\\_IGNORE\\_CONFIGURED\\_ENDPOINT\\_URLS](#)- Ignora l'endpoint URLs configurato.

Linux or macOS

```
$ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

Windows Command Prompt

Da impostare per tutte le sessioni

```
C:\> setx AWS_IGNORE_CONFIGURED_ENDPOINT_URLS true
```

Da impostare solo per la sessione corrente

```
C:\> set AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

## PowerShell

```
PS C:\> $Env:AWS_IGNORE_CONFIGURED_ENDPOINT_URLS="true"
```

- [AWS\\_ENDPOINT\\_URL](#)- Imposta un endpoint URL globale.

## Linux or macOS

```
$ export AWS_ENDPOINT_URL=http://localhost:4567
```

## Windows Command Prompt

Da impostare per tutte le sessioni

```
C:\> setx AWS_ENDPOINT_URL http://localhost:4567
```

Da impostare solo per la sessione corrente

```
C:\> set AWS_ENDPOINT_URL=http://localhost:4567
```

## PowerShell

```
PS C:\> $Env:AWS_ENDPOINT_URL="http://localhost:4567"
```

- Il config file:
  - [ignore\\_configure\\_endpoint\\_urls](#)- Ignora l'endpoint URLs configurato.

```
ignore_configure_endpoint_urls = true
```

- [endpoint\\_url](#)- Imposta un endpoint URL globale.

```
endpoint_url = http://localhost:4567
```

Gli endpoint specifici del servizio e l'opzione della `--endpoint-url` riga di comando sostituiscono qualsiasi endpoint globale.

## Imposta per utilizzare gli endpoint per tutti FIPs Servizi AWS

Per indirizzare le richieste relative a tutti i servizi all'utilizzo FIPs degli endpoint, utilizza uno dei seguenti metodi:

- [AWS\\_USE\\_FIPS\\_ENDPOINT](#) variabile di ambiente.

Linux or macOS

```
$ export AWS_USE_FIPS_ENDPOINT=true
```

Windows Command Prompt

Da impostare per tutte le sessioni

```
C:\> setx AWS_USE_FIPS_ENDPOINT true
```

Da impostare solo per la sessione corrente

```
C:\> set AWS_USE_FIPS_ENDPOINT=true
```

PowerShell

```
PS C:\> $Env:AWS_USE_FIPS_ENDPOINT="true"
```

- [use\\_fips\\_endpoint](#) impostazione del file.

```
use_fips_endpoint = true
```

Alcuni AWS servizi offrono endpoint che supportano il [Federal Information Processing Standard \(FIPS\) 140-2](#), in altri. Regioni AWS Se il AWS servizio supporta FIPS, questa impostazione specifica l'FIPS endpoint da utilizzare. AWS CLI A differenza degli AWS endpoint standard, gli FIPS endpoint utilizzano una libreria TLS software conforme a 140-2. FIPS Questi endpoint potrebbero essere necessari ad aziende che interagiscono con il governo degli Stati Uniti.

Se questa impostazione è abilitata, ma non esiste un FIPS endpoint per il servizio in uso Regione AWS, il comando potrebbe avere esito negativo. AWS In questo caso, specifica manualmente l'endpoint da utilizzare nel comando utilizzando l'[--endpoint-url](#) opzione o utilizza endpoint specifici del [servizio](#).

[Per ulteriori informazioni sulla specificazione degli endpoint in base al servizio, FIPS consulta Endpoints by Service Regione AWS. FIPS](#)

## Imposta per utilizzare endpoint dual-stack per tutti Servizi AWS

Per indirizzare le richieste relative a tutti i servizi in modo che utilizzino gli endpoint dual-stack quando disponibili, utilizza una delle seguenti impostazioni:

- [AWS\\_USE\\_DUALSTACK\\_ENDPOINT](#) variabile di ambiente.

Linux or macOS

```
$ export AWS_USE_DUALSTACK_ENDPOINT=true
```

Windows Command Prompt

Da impostare per tutte le sessioni

```
C:\> setx AWS_USE_DUALSTACK_ENDPOINT true
```

Da impostare solo per la sessione corrente

```
C:\> set AWS_USE_DUALSTACK_ENDPOINT=true
```

PowerShell

```
PS C:\> $Env:AWS_USE_DUALSTACK_ENDPOINT="true"
```

- [use\\_dualstack\\_endpoint](#) impostazione del file.

```
use_dualstack_endpoint = true
```

Consente l'uso di endpoint dual-stack per inviare richieste. AWS Per ulteriori informazioni sugli endpoint dual-stack, che supportano sia il traffico che il IPv6 traffico, consulta [Using IPv4 Using Amazon S3 dual-stack endpoint nella Amazon Simple Storage Service User Guide](#). Gli endpoint dual-stack sono disponibili per alcuni servizi in alcune regioni. Se non esiste un endpoint dual-stack per il servizio o se la richiesta ha esito negativo. Regione AWS Questa opzione è disabilitata per impostazione predefinita.



## Imposta endpoint specifici del servizio

La configurazione degli endpoint specifica per il servizio offre la possibilità di utilizzare un endpoint persistente di tua scelta per le richieste. AWS CLI Queste impostazioni offrono la flessibilità necessaria per supportare endpoint locali, VPC endpoint e ambienti di sviluppo locale di terze parti. AWS È possibile utilizzare endpoint diversi per ambienti di test e produzione. È possibile specificare un endpoint URL per singoli utenti. Servizi AWS

Gli endpoint specifici del servizio possono essere specificati nei seguenti modi:

- L'opzione della riga di comando [--endpoint-url](#) per un singolo comando.
- Variabili di ambiente:
  - [AWS\\_IGNORE\\_CONFIGURED\\_ENDPOINT\\_URLS](#)- Ignora tutti gli endpoint configuratiURLs, a meno che non sia specificato nella riga di comando.
  - [AWS\\_ENDPOINT\\_URL\\_<SERVICE>](#)- Specifica un endpoint personalizzato utilizzato per un servizio specifico, dove <SERVICE> viene sostituito con l'identificatore. Servizio AWS Per tutte le variabili specifiche del servizio, vedere. [the section called “Elenco di identificatori specifici del servizio”](#)
- configfile:
  - [ignore\\_configure\\_endpoint\\_urls](#)- Ignora tutti gli endpoint configuratiURLs, a meno che non sia specificato utilizzando le variabili di ambiente o sulla riga di comando.
  - La [services](#) sezione del config file combinata con l'impostazione del [endpoint\\_url](#) file.

Argomenti relativi agli endpoint specifici del servizio:

- [Endpoint specifici del servizio: variabili di ambiente](#)
- [Endpoint specifici del servizio: file condiviso config](#)
- [Endpoint specifici del servizio: elenco di identificatori specifici del servizio](#)

### Endpoint specifici del servizio: variabili di ambiente

Le variabili di ambiente sostituiscono le impostazioni nel file di configurazione, ma non sostituiscono le opzioni specificate nella riga di comando. Usa le variabili di ambiente se desideri che tutti i profili utilizzino gli stessi endpoint sul tuo dispositivo.

Le seguenti sono variabili di ambiente specifiche del servizio:

- [AWS\\_IGNORE\\_CONFIGURED\\_ENDPOINT\\_URLS](#)- Ignora tutti gli endpoint configurati URLs, a meno che non sia specificato nella riga di comando.

Linux or macOS

```
$ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

Windows Command Prompt

Da impostare per tutte le sessioni

```
C:\> setx AWS_IGNORE_CONFIGURED_ENDPOINT_URLS true
```

Da impostare solo per la sessione corrente

```
C:\> set AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
```

PowerShell

```
PS C:\> $Env:AWS_IGNORE_CONFIGURED_ENDPOINT_URLS="true"
```

- [AWS\\_ENDPOINT\\_URL\\_<SERVICE>](#)- Specifica un endpoint personalizzato utilizzato per un servizio specifico, dove <SERVICE> viene sostituito con l' Servizio AWS identificatore. Per tutte le variabili specifiche del servizio, vedere. [the section called "Elenco di identificatori specifici del servizio"](#)

I seguenti esempi di variabili di ambiente impostano un endpoint per: AWS Elastic Beanstalk

Linux or macOS

```
$ export AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:4567
```

Windows Command Prompt

Da impostare per tutte le sessioni

```
C:\> setx AWS_ENDPOINT_URL_ELASTIC_BEANSTALK http://localhost:4567
```

Da impostare solo per la sessione corrente

```
C:\> set AWS_ENDPOINT_URL_ELASTIC_BEANSTALK=http://localhost:4567
```

## PowerShell

```
PS C:\> $Env:AWS_ENDPOINT_URL_ELASTIC_BEANSTALK="http://localhost:4567"
```

Per ulteriori informazioni sull'impostazione delle variabili di ambiente, vedere [the section called "Variabili di ambiente"](#).

## Endpoint specifici del servizio: file condiviso **config**

Nel `config` file condiviso, `endpoint_url` viene utilizzato in più sezioni. Per impostare un endpoint specifico del servizio, utilizza l'`endpoint_url` impostazione annidata sotto una chiave identificativa del servizio all'interno di una sezione. `services` Per i dettagli sulla definizione di una `services` sezione nel file condiviso, consulta. `config` [the section called "services"](#)

L'esempio seguente utilizza una `services` sezione per configurare un endpoint specifico del servizio per Amazon URL S3 e un endpoint globale personalizzato utilizzato per tutti gli altri servizi:

```
[profile dev1]  
endpoint_url = http://localhost:1234  
services = s3-specific  
  
[services testing-s3]  
s3 =  
  endpoint_url = http://localhost:4567
```

Un singolo profilo può configurare gli endpoint per più servizi. L'esempio seguente imposta l'endpoint specifico del servizio per Amazon URLs S3 e AWS Elastic Beanstalk nello stesso profilo.

Per un elenco di tutte le chiavi identificative del servizio da utilizzare nella sezione, consulta. `services` [Elenco di identificatori specifici del servizio](#)

```
[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
```

La sezione di configurazione del servizio può essere utilizzata in più profili. L'esempio seguente presenta due profili che utilizzano la stessa `services` definizione:

```
[profile dev1]
output = json
services = testing-s3

[profile dev2]
output = text
services = testing-s3

[services testing-s3]
s3 =
  endpoint_url = https://localhost:4567
```

## Endpoint specifici del servizio: elenco di identificatori specifici del servizio

L' Servizio AWS identificatore si basa su quello del API modello e sostituisce tutti gli spazi con caratteri `serviceId` di sottolineatura e tutte le lettere minuscole.

Il seguente esempio di identificatore di servizio utilizza. AWS Elastic Beanstalk AWS Elastic Beanstalk ha un `serviceId` of [Elastic Beanstalk](#), quindi la chiave identificativa del servizio è. `elastic_beanstalk`

La tabella seguente elenca tutti gli identificatori, le chiavi di `config` file e le variabili di ambiente specifici del servizio.

## Priorità della configurazione e delle impostazioni degli endpoint

Le impostazioni di configurazione degli endpoint si trovano in più posizioni, ad esempio nelle variabili di sistema o di ambiente utente, nei file di AWS configurazione locali, oppure sono dichiarate esplicitamente nella riga di comando come parametro. Le impostazioni di configurazione AWS CLI degli endpoint hanno la precedenza nell'ordine seguente:

1. L'opzione della `--endpoint-url` riga di comando.
2. Se abilitata, la variabile di ambiente `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` globale dell'endpoint o l'impostazione del profilo `ignore_configure_endpoint_urls` per ignorare gli endpoint personalizzati.
3. Il valore fornito da una variabile `AWS_ENDPOINT_URL_<SERVICE>` di ambiente specifica del servizio, ad esempio. `AWS_ENDPOINT_URL_DYNAMODB`

4. I valori forniti dalle variabili [AWS\\_USE\\_DUALSTACK\\_ENDPOINT](#), [AWS\\_USE\\_FIPS\\_ENDPOINT](#), e di [AWS\\_ENDPOINT\\_URL](#) ambiente.
5. Il valore dell'endpoint specifico del servizio fornito dall'[endpoint\\_url](#) impostazione all'interno di una `services` sezione del file condiviso. `config`
6. Il valore fornito dall'[endpoint\\_url](#) impostazione all'interno di uno `profile` dei file condivisi. `config`
7. [use\\_dualstack\\_endpoint](#), [use\\_fips\\_endpoint](#) e [endpoint\\_url](#) impostazioni.
8. Qualsiasi endpoint predefinito URL per il rispettivo dispositivo Servizio AWS viene utilizzato per ultimo. Per un elenco degli endpoint del servizio standard disponibili in ogni regione, consulta [Regioni ed endpoint AWS](#) nella Riferimenti generali di Amazon Web Services.

# Credenziali di autenticazione e accesso per AWS CLI

È necessario stabilire la modalità di AWS CLI autenticazione AWS quando si sviluppa con AWS i servizi. Per configurare le credenziali per l'accesso programmatico per il AWS CLI, scegliete una delle seguenti opzioni. Le opzioni sono in ordine di raccomandazione.

Quale utente necessita dell'accesso programmatico?	Scopo	Istruzioni
IAM	Usa a breve termine.	<a href="#">the section called “Credenziali a breve termine”</a>
IAM	Usa i ruoli come credenziali.	<a href="#">the section called “IAMruoli”</a>
IAM	(Non consigliato) Utilizza credenziali a lungo termine.	<a href="#">the section called “IAMutenti”</a>

## Configurazione e precedenza delle credenziali

Le credenziali e le impostazioni di configurazione si trovano in più posizioni, ad esempio le variabili di sistema o di ambiente utente, i file di AWS configurazione locali, oppure sono dichiarate esplicitamente nella riga di comando come parametro. Alcune posizioni hanno la precedenza su altre. Le credenziali e le impostazioni di configurazione AWS CLI hanno la precedenza nell'ordine seguente:

1. [Opzioni della riga di comando](#): sostituisce le impostazioni in qualsiasi altra posizione, ad esempio i `--region` parametri e `--output --profile`
2. [Variabili di ambiente](#): è possibile memorizzare valori nelle variabili di ambiente del sistema.
3. [Assumi un ruolo](#): assumi le autorizzazioni di un IAM ruolo tramite la configurazione o il `aws sts assume-role` comando.
4. [Assumi il ruolo con l'identità web](#): assumi le autorizzazioni di un IAM ruolo utilizzando l'identità web tramite la configurazione o il `aws sts assume-role` comando.
5. [File delle credenziali](#): il config file `credentials` and viene aggiornato quando si esegue il comando `aws configure` Il `credentials` file si trova `~/.aws/credentials` in Linux o macOS o `C:\Users\USERNAME\.aws\credentials` in Windows.

6. [Processo personalizzato](#): ottieni le tue credenziali da una fonte esterna.
7. [File di configurazione](#): il config file `credentials` and viene aggiornato quando si esegue il comando `aws configure`. Il config file si trova `~/.aws/config` in Linux o macOS o `C:\Users\USERNAME\.aws\config` in Windows.
8. [Credenziali del contenitore](#): puoi associare un IAM ruolo a ciascuna delle definizioni di attività di Amazon Elastic Container Service (AmazonECS). Le credenziali temporanee per tale ruolo sono quindi disponibili per i contenitori dell'attività. Per ulteriori informazioni, consulta [IAMRoles for Tasks](#) nella Amazon Elastic Container Service Developer Guide.
9. [Credenziali del profilo dell'EC2istanza Amazon](#): puoi associare un IAM ruolo a ciascuna delle tue istanze Amazon Elastic Compute Cloud EC2 (Amazon). Le credenziali temporanee per tale ruolo sono quindi disponibili per il codice in esecuzione nell'istanza. Le credenziali vengono fornite tramite il servizio di EC2 metadati Amazon. Per ulteriori informazioni, consulta [IAMRoles for Amazon EC2 nella Amazon EC2 User Guide](#) e [Using Instance Profiles](#) nella IAMUser Guide.

## Argomenti aggiuntivi in questa sezione

- [the section called “Credenziali a breve termine”](#)
- [the section called “IAMruoli”](#)
- [the section called “IAMutenti”](#)
- [the section called “Utilizzo dei metadati delle EC2 istanze Amazon come credenziali nel AWS CLI”](#)
- [the section called “Credenziali esterne”](#)

## Autenticazione con credenziali a breve termine per AWS CLI

Ti consigliamo di configurare il tuo strumento SDK o il tuo strumento per utilizzare l'[autenticazione IAM Identity Center](#) con opzioni di durata estesa della sessione. Tuttavia, è possibile copiare e utilizzare le credenziali temporanee disponibili nel portale di AWS accesso. Le nuove credenziali dovranno essere copiate quando scadono. È possibile utilizzare le credenziali temporanee in un profilo o utilizzarle come valori per le proprietà di sistema e le variabili di ambiente.

1. [Accedi al portale di AWS accesso](#).
2. Segui [queste istruzioni](#) per copiare le credenziali del IAM ruolo dal portale di AWS accesso.





# Utilizzo di un IAM ruolo in AWS CLI

Un [ruolo AWS Identity and Access Management \(IAM\)](#) è uno strumento di autorizzazione che consente a un utente di ottenere autorizzazioni aggiuntive (o diverse) o di ottenere le autorizzazioni per eseguire azioni in un account diverso AWS .

## Argomenti

- [Prerequisiti](#)
- [Panoramica sull'utilizzo dei IAM ruoli](#)
- [Configurazione e utilizzo di un ruolo](#)
- [Utilizzo dell'autenticazione a più fattori](#)
- [Ruoli per più account e ID esterno](#)
- [Specifica di un nome di sessione del ruolo per semplificare l'audit](#)
- [Assunzione di un ruolo con un'identità Web](#)
- [Cancellazione delle credenziali nella cache](#)

## Prerequisiti

Per eseguire i iam comandi, è necessario installare e configurare. AWS CLI Per ulteriori informazioni, consulta [Installare il AWS CLI](#).

## Panoramica sull'utilizzo dei IAM ruoli

È possibile configurare AWS Command Line Interface (AWS CLI) per l'utilizzo di un IAM ruolo definendo un profilo per il ruolo nel `~/.aws/config` file.

L'esempio seguente mostra un profilo del ruolo denominato `marketingadmin`. Se esegui comandi con `--profile marketingadmin` (o li specifichi con la [variabile di AWS\\_PROFILE ambiente](#)), AWS CLI utilizza le credenziali definite in un profilo separato `user1` per assumere il ruolo con Amazon Resource Name (ARN)`arn:aws:iam::123456789012:role/marketingadminrole`. Puoi eseguire tutte le operazioni consentite dalle autorizzazioni assegnate a tale ruolo.

```
[profile marketingadmin]  
role_arn = arn:aws:iam::123456789012:role/marketingadminrole  
source_profile = user1
```

Puoi quindi specificare un profilo con nome separato `source_profile` che contenga le credenziali utente con l'autorizzazione a utilizzare il ruolo. Nell'esempio precedente il profilo `marketingadmin` utilizza le credenziali del profilo `user1`. Quando si specifica che un AWS CLI comando deve utilizzare il profilo `marketingadmin`, cerca AWS CLI automaticamente le credenziali per il `user1` profilo collegato e le utilizza per richiedere credenziali temporanee per il ruolo specificato. IAM A tale scopo CLI utilizza l'AssumeRoleoperazione [sts:](#) in background. Queste credenziali temporanee vengono quindi utilizzate per eseguire il comando richiesto AWS CLI . Al ruolo specificato devono essere associate politiche di IAM autorizzazione che consentano l'esecuzione del AWS CLI comando richiesto.

Per eseguire un AWS CLI comando da un'istanza Amazon Elastic Compute Cloud (AmazonEC2) o da un contenitore Amazon Elastic Container Service (AmazonECS), puoi utilizzare un IAM ruolo collegato al profilo dell'istanza o al contenitore. Se non specifichi alcun profilo o non imposti variabili di ambiente, tale ruolo viene utilizzato direttamente. In questo modo, potrai evitare di archiviare chiavi di accesso di lunga durata nelle istanze. Puoi anche utilizzare i ruoli dell'istanza o del container solo per ottenere le credenziali per un altro ruolo. A tale scopo, usa `credential_source` (anziché `source_profile`) per specificare come trovare le credenziali. L'attributo `credential_source` supporta i seguenti valori:

- `Environment`— Recupera le credenziali di origine dalle variabili di ambiente.
- `Ec2InstanceMetadata`— Utilizza il IAM ruolo associato al profilo dell'EC2istanza Amazon.
- `EcsContainer`— Utilizza il IAM ruolo associato al ECS contenitore Amazon.

L'esempio seguente mostra lo stesso `marketingadminrole` ruolo utilizzato facendo riferimento a un profilo di EC2 istanza Amazon.

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
credential_source = Ec2InstanceMetadata
```

Quando si richiama un ruolo, sono disponibili opzioni aggiuntive che è possibile richiedere, ad esempio l'utilizzo dell'autenticazione a più fattori e di un ID esterno (utilizzato da società terze per accedere alle risorse dei client). Puoi anche specificare nomi di sessione di ruolo univoci che possono essere controllati più facilmente nei AWS CloudTrail log.

## Configurazione e utilizzo di un ruolo

Quando esegui comandi utilizzando un profilo che specifica un IAM ruolo, AWS CLI utilizza le credenziali del profilo di origine per chiamare AWS Security Token Service (AWS STS) e richiedere credenziali temporanee per il ruolo specificato. L'utente nel profilo di origine deve disporre dell'autorizzazione per chiamare `sts:assume-role` per il ruolo nel profilo specificato. Il ruolo deve disporre di una relazione di trust che consente all'utente nel profilo di origine di usare il ruolo. Il processo di recupero e quindi di utilizzo delle credenziali temporanee per un ruolo è spesso definito come assunzione del ruolo.

Puoi creare un ruolo IAM con le autorizzazioni che desideri che gli utenti assumano seguendo la procedura riportata in [Creazione di un ruolo per delegare le autorizzazioni a un utente nella Guida per l'utente](#). IAM AWS Identity and Access Management Se il ruolo e l'utente del profilo di origine appartengono allo stesso account, puoi immettere il tuo ID account durante la configurazione della relazione di trust del ruolo.

Dopo aver creato il ruolo, modifica la relazione di fiducia per consentire all'utente di assumerlo.

L'esempio seguente mostra una policy di trust che può essere collegata a un ruolo. Questa politica consente a qualsiasi utente dell'account 123456789012 di assumere il ruolo, se l'amministratore di tale account concede esplicitamente l'autorizzazione all'utente. `sts:AssumeRole`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

La policy di trust non concede attualmente autorizzazioni. L'amministratore dell'account deve delegare l'autorizzazione per assumere il ruolo a singoli utenti collegando una policy con le autorizzazioni appropriate. L'esempio seguente mostra una politica che è possibile allegare a un utente che consente all'utente di assumere solo il ruolo. `marketingadminrole` Per ulteriori

informazioni sulla concessione a un utente dell'accesso per assumere un ruolo, vedere [Concessione dell'autorizzazione di un utente a cambiare ruolo nella Guida](#) per l'IAMutente.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::123456789012:role/marketingadminrole"
    }
  ]
}
```

L'utente non deve disporre di autorizzazioni aggiuntive per eseguire AWS CLI i comandi utilizzando il profilo del ruolo. Al contrario, le autorizzazioni per eseguire il comando provengono da quelle associate al ruolo. Si allegano politiche di autorizzazione al ruolo per specificare quali azioni possono essere eseguite su quali AWS risorse. Per ulteriori informazioni sull'assegnazione delle autorizzazioni a un ruolo (che funziona in modo identico a un utente), consulta [Modifica delle autorizzazioni per un IAM utente nella Guida per l'utente](#). IAM

Ora che il profilo del ruolo, le autorizzazioni del ruolo, la relazione di trust del ruolo e le autorizzazioni utente sono configurate correttamente, puoi utilizzare il ruolo nella riga di comando richiamando l'opzione `--profile`. Ad esempio, quanto segue chiama il `ls` comando Amazon S3 utilizzando le autorizzazioni associate al `marketingadmin` ruolo come definito nell'esempio all'inizio di questo argomento.

```
$ aws s3 ls --profile marketingadmin
```

Per utilizzare il ruolo per diverse chiamate, puoi impostare la variabile di ambiente `AWS_PROFILE` per la sessione corrente dalla riga di comando. Sebbene la variabile di ambiente sia definita, non è necessario specificare l'opzione `--profile` su ogni comando.

Linux o macOS

```
$ export AWS_PROFILE=marketingadmin
```

Windows

```
C:\> setx AWS_PROFILE marketingadmin
```

Per ulteriori informazioni sulla configurazione di utenti e ruoli, consulta [IAMIdentità \(utenti, gruppi di utenti e ruoli\)](#) e [IAMruoli nella Guida per l'IAMutente](#).

## Utilizzo dell'autenticazione a più fattori

Per una maggiore sicurezza, puoi richiedere che gli utenti forniscano una chiave monouso generata da un dispositivo di autenticazione a più fattori (MFA), un dispositivo U2F o un'app mobile quando tentano di effettuare una chiamata utilizzando il profilo del ruolo.

Innanzitutto, puoi scegliere di modificare la relazione di fiducia in base al ruolo da richiedere. IAM MFA Ciò impedisce a chiunque di utilizzare il ruolo senza prima autenticarsi utilizzando MFA. Per un esempio, consulta la riga `Condition` nel seguente esempio. Questa politica consente all'utente denominato `anika` assumere il ruolo a cui è associata la politica, ma solo se si autentica utilizzando 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 } }
    }
  ]
}
```

Quindi, aggiungi una riga al profilo del ruolo che specifica il ARN dispositivo dell'MFAutente. Le seguenti voci di config file di esempio mostrano due profili di ruolo che utilizzano entrambi le chiavi di accesso per consentire `anika` all'utente di richiedere credenziali temporanee per il ruolo. `cli-role` L'utente `anika` ha le autorizzazioni per assumere quel ruolo concesse dalla policy di trust del ruolo.

```
[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
```

L'`mfa_serial` impostazione può richiedere ARN, come illustrato, il numero di serie di un MFA token hardware.

Il primo profilo non richiede MFA. `role-without-mfa` Tuttavia, poiché la policy di fiducia dell'esempio precedente allegata al ruolo richiede MFA, qualsiasi tentativo di eseguire un comando con questo profilo ha esito negativo.

```
$ aws iam list-users --profile role-without-mfa
```

```
An error occurred (AccessDenied) when calling the AssumeRole operation: Access denied
```

La seconda voce del profilo `role-with-mfa`, identifica un MFA dispositivo da utilizzare. Quando l'utente tenta di eseguire un AWS CLI comando con questo profilo, AWS CLI richiede all'utente di inserire la password monouso (OTP) fornita dal dispositivo. MFA Se l'MFA autenticazione ha esito positivo, il comando esegue l'operazione richiesta. Non OTP viene visualizzato sullo schermo.

```
$ aws iam list-users --profile role-with-mfa
Enter MFA code for arn:aws:iam::123456789012:mfa/cli-user:
{
  "Users": [
    {
      ...
```

## Ruoli per più account e ID esterno

Puoi abilitare gli utenti in modo che assumano ruoli appartenenti ad account diversi configurando il ruolo come ruolo per più account. Durante la creazione del ruolo, imposta il tipo di ruolo su Un altro AWS account, come descritto in [Creazione di un ruolo per delegare le autorizzazioni a un IAM utente](#).

Facoltativamente, seleziona Richiedi. MFA MFARequire configura la condizione appropriata nella relazione di trust, come descritto in. [Utilizzo dell'autenticazione a più fattori](#)

Se utilizzi un [ID esterno](#) per fornire ulteriore controllo su chi può usare un ruolo per diversi account, devi aggiungere il parametro `external_id` al profilo del ruolo. Ciò viene utilizzato in genere solo quando l'altro account è controllato da qualcuno esterno all'azienda o organizzazione.

```
[profile crossaccountrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
mfa_serial = arn:aws:iam::123456789012:mfa/saanvi
external_id = 123456
```

## Specifica di un nome di sessione del ruolo per semplificare l'audit

Quando molti individui condividono un ruolo, l'audit diventa più difficile. Si supponga che tu voglia associare ogni operazione richiamata alla persona che ha richiamato l'operazione. Tuttavia, quando la persona utilizza un ruolo, l'assunzione del ruolo da parte dell'individuo è un'operazione distinta rispetto alla chiamata di un'operazione ed è pertanto necessario correlare manualmente le due cose.

Puoi semplificare questa operazione specificando nomi univoci delle sessioni del ruolo quando gli utenti assumono un ruolo. A tale scopo, aggiungi un parametro `role_session_name` a ogni profilo denominato nel file config che specifica un ruolo. Il `role_session_name` valore viene passato all'AssumeRoleoperazione e diventa parte della sessione ARN per il ruolo. È inoltre incluso nei AWS CloudTrail registri di tutte le operazioni registrate.

Ad esempio, puoi creare un profilo basato su ruoli come segue.

```
[profile namedsessionrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
role_session_name = Session_Maria_Garcia
```

Ciò fa sì che la sessione di ruolo abbia quanto segue. ARN

```
arn:aws:iam::234567890123:assumed-role/SomeRole/Session_Maria_Garcia
```

Inoltre, tutti i AWS CloudTrail log includono il nome della sessione di ruolo nelle informazioni acquisite per ogni operazione.

## Assunzione di un ruolo con un'identità Web

È possibile configurare un profilo per indicare che AWS CLI devono assumere un ruolo utilizzando la [federazione delle identità Web e Open ID Connect \(OIDC\)](#). Quando lo specificate in un profilo, effettua AWS CLI automaticamente la AWS STS `AssumeRoleWithWebIdentity` chiamata corrispondente per voi.

### Note

Quando si specifica un profilo che utilizza un IAM ruolo, AWS CLI effettua le chiamate appropriate per recuperare le credenziali temporanee. Queste credenziali sono memorizzate in `~/.aws/cli/cache`. AWS CLI I comandi successivi che specificano lo stesso profilo utilizzano le credenziali temporanee memorizzate nella cache fino alla scadenza. A quel punto, aggiorna AWS CLI automaticamente le credenziali.

Per recuperare e utilizzare le credenziali temporanee utilizzando la federazione delle identità Web, puoi specificare i seguenti valori di configurazione in un profilo condiviso.

### [role\\_arn](#)

Specifica il ruolo ARN da assumere.

### `web_identity_token_file`

Specifica il percorso di un file che contiene un token di accesso OAuth 2.0 o un token OpenID Connect ID fornito dal provider di identità. L' AWS CLI carica questo file e passa il relativo contenuto come argomento `WebIdentityToken` all'operazione `AssumeRoleWithWebIdentity`.

### [role\\_session\\_name](#)

Specifica un nome facoltativo applicato a questa sessione `assume-role`.

Di seguito è riportato un esempio di configurazione per la quantità minima di configurazione necessaria per configurare un profilo del ruolo assunto con un'identità Web:

```
# In ~/.aws/config

[profile web-identity]
role_arn=arn:aws:iam:123456789012:role/RoLeNameToAssume
```



```
web_identity_token_file=/path/to/a/token
```

Puoi anche fornire questa configurazione utilizzando [variabili di ambiente](#):

**AWS\_ROLE\_ARN**

Il ARN ruolo da assumere.

**AWS\_WEB\_IDENTITY\_TOKEN\_FILE**

Il percorso del file del token dell'identità Web.

**AWS\_ROLE\_SESSION\_NAME**

Il nome applicato a questa sessione assume-role.

#### Note

Queste variabili di ambiente attualmente si applicano solo al ruolo assumere con provider di identità Web. Non si applicano alla configurazione generale del provider dell'assunzione di ruoli.

## Cancellazione delle credenziali nella cache

Quando si utilizza un ruolo, le credenziali temporanee vengono memorizzate nella AWS CLI cache locale fino alla loro scadenza. La prossima volta che provi a utilizzarle, AWS CLI tenta di rinnovarle per tuo conto.

Se le credenziali temporanee del ruolo vengono [revoke](#), non vengono rinnovate automaticamente e i tentativi di utilizzarle non vanno a buon fine. Tuttavia, puoi eliminare la cache per AWS CLI forzare il recupero di nuove credenziali.

Linux o macOS

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\aws\cli\cache
```

# Autenticazione tramite credenziali IAM utente per AWS CLI

## Warning

Per evitare rischi per la sicurezza, non utilizzate IAM gli utenti per l'autenticazione quando sviluppate software appositamente creato o lavorate con dati reali. Utilizza invece la federazione con un provider di identità come [AWS IAM Identity Center](#).

Questa sezione spiega come configurare le impostazioni di base con un utente. IAM Queste includono le credenziali di sicurezza che utilizzano i `credentials` file `config` and.

## Argomenti

- [Fase 1: Crea il tuo IAM utente](#)
- [Passaggio 2: ottieni le tue chiavi di accesso](#)
- [Configura il AWS CLI](#)
  - [Uso di `aws configure`](#)

## Fase 1: Crea il tuo IAM utente

Crea il tuo IAM utente seguendo la procedura [Creazione di IAM utenti \(console\)](#) nella Guida per l'IAMutente.

- Per le opzioni di autorizzazione, scegli `Allega` direttamente le politiche in base al modo in cui desideri assegnare le autorizzazioni a questo utente.
- La maggior parte dei SDK tutorial «Guida introduttiva» utilizza il servizio Amazon S3 come esempio. Per fornire alla tua applicazione l'accesso completo ad Amazon S3, seleziona la `AmazonS3FullAccess` policy da allegare a questo utente.

## Passaggio 2: ottieni le tue chiavi di accesso

1. Accedi a AWS Management Console e apri la IAM console all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione della IAM console, seleziona `Utenti`, quindi seleziona `User` **name**l'utente che hai creato in precedenza.

3. Nella pagina dell'utente, seleziona la pagina Credenziali di sicurezza. Quindi, in Chiavi di accesso, seleziona Crea chiave di accesso.
4. Per Creare una chiave di accesso (Fase 1), scegliete Command Line Interface (CLI).
5. Per la creazione della chiave di accesso (Fase 2), inserite un tag opzionale e selezionate Avanti.
6. Per il passaggio 3 della creazione della chiave di accesso, seleziona Scarica il file.csv per salvare un .csv file con la chiave di accesso e la chiave di accesso segreta IAM dell'utente. Queste informazioni ti serviranno per dopo.
7. Seleziona Done (Fatto)

## Configura il AWS CLI

Per uso generale, sono AWS CLI necessarie le seguenti informazioni:

- ID chiave di accesso
- Chiave di accesso segreta
- AWS Regione
- Formato di output

AWS CLI Memorizza queste informazioni in un profilo (una raccolta di impostazioni) denominato `default` nel `credentials` file. Per impostazione predefinita, le informazioni in questo profilo vengono utilizzate quando si esegue un AWS CLI comando che non specifica esplicitamente un profilo da utilizzare. Per ulteriori informazioni sul file `credentials`, consulta [Impostazioni dei file di configurazione e credenziali in AWS CLI](#).

Per configurare AWS CLI, utilizzare una delle seguenti procedure:

Argomenti

- [Uso di `aws configure`](#)

### Uso di `aws configure`

Per uso generale, il `aws configure` comando è il modo più veloce per configurare AWS CLI l'installazione. Questa procedura guidata di configurazione richiede ogni informazione necessaria per iniziare. Se non diversamente specificato utilizzando l'`--profile` opzione, AWS CLI memorizza queste informazioni nel profilo `default`

L'esempio seguente configura un default profilo utilizzando valori di esempio. Sostituirli con i propri valori come descritto nelle sezioni seguenti.

```
$ 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
```

L'esempio seguente configura un profilo denominato `userprod` utilizzando valori di esempio. Sostituirli con i propri valori come descritto nelle sezioni seguenti.

```
$ 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
```

## Utilizzo dei metadati delle EC2 istanze Amazon come credenziali nel AWS CLI

Quando esegui il comando AWS CLI da un'istanza Amazon Elastic Compute Cloud (AmazonEC2), puoi semplificare l'immissione delle credenziali ai comandi. Ogni EC2 istanza Amazon contiene metadati che AWS CLI possono essere interrogati direttamente per ottenere credenziali temporanee. Quando un IAM ruolo è associato all'istanza, recupera AWS CLI automaticamente e in modo sicuro le credenziali dai metadati dell'istanza.

Per disabilitare questo servizio, utilizzate la variabile di ambiente [AWS\\_ECMETADATA2](#) `DISABLED`.

### Argomenti

- [Prerequisiti](#)
- [Configurazione di un profilo per i metadati Amazon EC2](#)

## Prerequisiti

Per utilizzare EC2 le credenziali Amazon con AWS CLI, devi completare quanto segue:

- Installa e configura la AWS CLI. Per ulteriori informazioni, consulta [Installare il AWS CLI](#) e [Credenziali di autenticazione e accesso per AWS CLI](#).
- Conosci i file di configurazione e i profili denominati. Per ulteriori informazioni, consulta [Impostazioni dei file di configurazione e credenziali in AWS CLI](#).
- Hai creato un ruolo AWS Identity and Access Management (IAM) che ha accesso alle risorse necessarie e lo hai associato all'EC2istanza Amazon al momento del lancio. Per ulteriori informazioni, consulta [IAMle politiche di Amazon EC2 nella Amazon EC2 User Guide](#) e [Concedere alle applicazioni eseguite su Amazon EC2 Instances l'accesso alle AWS risorse](#) nella Guida per l'IAMutente.

## Configurazione di un profilo per i metadati Amazon EC2

Per specificare che desideri utilizzare le credenziali disponibili nel profilo dell'EC2istanza Amazon di hosting, utilizza la seguente sintassi nel profilo denominato nel tuo file di configurazione. Per ulteriori istruzioni, consulta i passaggi seguenti.

```
[profile profilename]  
role_arn = arn:aws:iam::123456789012:role/rolename  
credential_source = Ec2InstanceMetadata  
region = region
```

1. Crea un profilo nel tuo file di configurazione.

```
[profile profilename]
```

2. Aggiungi il tuo ruolo IAM arn che ha accesso alle risorse necessarie.

```
role_arn = arn:aws:iam::123456789012:role/rolename
```

3. Specificalo Ec2InstanceMetadata come fonte delle credenziali.

```
credential_source = Ec2InstanceMetadata
```

4. Imposta la tua regione.

```
region = region
```

## Esempio

L'esempio seguente presuppone il *ruolo di amministratore di marketing* ruolo e utilizza la *us-west-2* regione in un profilo di EC2 istanza Amazon denominato *marketingadmin*.

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
credential_source = Ec2InstanceMetadata
region = us-west-2
```

## Acquisizione di credenziali con un processo esterno in AWS CLI

### Warning

Il seguente argomento illustra le credenziali di approvvigionamento da un processo esterno. Potrebbe insorgere un rischio per la sicurezza se il comando per generare le credenziali diventa accessibile a utenti o processi non riconosciuti. Ti consigliamo di utilizzare le alternative sicure e supportate fornite da AWS CLI e AWS per ridurre il rischio di compromissione delle tue credenziali. Assicurati di proteggere il file `config` e altri strumenti e file di supporto per evitare la divulgazione.

Assicurati che il tuo strumento personalizzato per le credenziali non scriva informazioni segrete, `StdErr` poiché è in AWS CLI grado di acquisire SDKs e registrare tali informazioni, esponendole potenzialmente a utenti non autorizzati.

Se disponi di un metodo per generare o cercare credenziali che non è direttamente supportato da AWS CLI, puoi configurarlo AWS CLI per utilizzarlo configurando `credential_process` impostazione nel file `config`

Puoi ad esempio includere una voce simile al seguente file `config`.

```
[profile developer]
credential_process = /opt/bin/awscreds-custom --username helen
```

## Sintassi

Per creare questa stringa in modo compatibile con qualsiasi sistema operativo, segui queste regole:

- Se il percorso o il nome del file contiene uno spazio, circondare il percorso completo e il nome del file con virgolette doppie (" "). Il percorso e il nome del file possono essere composti solo dai caratteri: A-Z a-z 0-9 - \_ . spazio
- Se un nome di parametro o un valore di parametro contiene uno spazio, circondare tale elemento con virgolette doppie (" "). È possibile racchiudere solo il nome o il valore, non la coppia.
- Non includere variabili di ambiente nelle stringhe. Ad esempio, non puoi includere \$HOME o %USERPROFILE%.
- Non specificare la cartella home come ~. Devi specificare il percorso completo.

### Esempio per Windows

```
credential_process = "C:\Path\To\credentials.cmd" parameterWithoutSpaces "parameter with spaces"
```

### Esempio per Linux o macOS

```
credential_process = "/Users/Dave/path/to/credentials.sh" parameterWithoutSpaces "parameter with spaces"
```

### Output previsto dal programma Credentials


AWS CLI Esegue il comando come specificato nel profilo e quindi legge i dati da. STDOUT Il comando specificato deve generare un JSON output STDOUT che corrisponda alla sintassi seguente.

```
{  
  "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

Al momento della stesura del presente documento, la chiave `Version` deve essere configurata su 1. Questo valore potrebbe incrementare nel tempo, man mano che la struttura evolve.

La `Expiration` chiave è un timestamp in formato [ISO8601](#). Se la `Expiration` chiave non è presente nell'output dello strumento, si CLI presuppone che le credenziali siano credenziali a lungo termine che non si aggiornano. In caso contrario, le credenziali vengono considerate provvisorie e aggiornate automaticamente mediante una nuova esecuzione del comando `credential_process` prima della relativa scadenza.

 Note

AWS CLI Non memorizza nella cache le credenziali dei processi esterni nel modo in cui presuppone le credenziali di ruolo. Se il caching è necessario, dovrai implementarlo nel processo esterno.

Il processo esterno può restituire un codice diverso da zero per indicare che si è verificato un errore durante il recupero delle credenziali.



# Utilizzando il AWS CLI

Questa sezione fornisce una panoramica completa dell'uso generale, delle funzionalità comuni e delle opzioni disponibili in AWS Command Line Interface (AWS CLI), oltre ai dettagli trattati nella [the section called "Endpoints"](#) sezione Configurazione.

Questa guida approfondisce gli aspetti fondamentali della scrittura dei AWS CLI comandi, incluse la struttura di base, la formattazione e le funzionalità di filtraggio. Comprendendo questi elementi fondamentali, sarete in grado di creare comandi che indirizzano con precisione le risorse e le azioni necessarie, senza la necessità di navigare su complesse console basate sul Web.

Inoltre, questo evidenzia il contenuto della guida e la documentazione disponibili per. AWS CLI Dalla guida a riga di comando integrata alla [guida di AWS CLI riferimento](#) completa alla , avrai accesso alle informazioni per aiutarti a esplorare le caratteristiche e le funzionalità di AWS CLI.

Per esempi e casi d'uso Servizio AWS specifici, consulta la [guida AWS CLI di riferimento Esempi di codice o la guida](#) . Questi forniscono informazioni specifiche sui comandi e mostrano esempi su come sfruttarli AWS CLI per vari Servizi AWS scopi.

## Note

Per impostazione predefinita, AWS CLI invia le richieste a Servizi AWS utilizzando HTTPS la TCP porta 443. Per garantire il corretto utilizzo di AWS CLI, è necessario essere in grado di effettuare connessioni in uscita su questa porta.

## Argomenti di questa guida

- [Accesso all'aiuto e alle risorse per AWS CLI](#)
- [Struttura di comando in AWS CLI](#)
- [Specificazione dei valori dei parametri nel AWS CLI](#)
- [Controllo dell'output dei comandi in AWS CLI](#)
- [Codici restituiti dalla riga di comando in AWS CLI](#)
- [Creazione e utilizzo di alias in AWS CLI](#)

# Accesso all'aiuto e alle risorse per AWS CLI

Questo argomento descrive come accedere ai contenuti di aiuto per AWS Command Line Interface (AWS CLI).

## Argomenti

- [Il comando AWS CLI help integrato](#)
- [AWS CLI guida di riferimento](#)
- [API documentazione](#)
- [Risoluzione degli errori](#)
- [Aiuto aggiuntivo](#)

## Il comando AWS CLI help integrato

È possibile ottenere assistenza con qualsiasi comando quando si utilizza il comando AWS Command Line Interface (AWS CLI). Per farlo, basta digitare `help` al termine del nome di un comando.

Ad esempio, il comando seguente mostra la guida per le AWS CLI opzioni generali e i comandi di primo livello disponibili.

```
$ aws help
```

Il comando seguente mostra i comandi specifici di Amazon Elastic Compute Cloud (AmazonEC2) disponibili.

```
$ aws ec2 help
```

L'esempio seguente mostra una guida dettagliata per il EC2 `DescribeInstances` funzionamento di Amazon. L'assistenza include le descrizioni dei parametri di input, i filtri disponibili e tutto ciò che è incluso nell'output. Include anche esempi che mostrano come digitare varianti comuni del comando.

```
$ aws ec2 describe-instances help
```

Le informazioni della Guida per ciascun comando sono suddivise in sei sezioni:

### Nome

Nome del comando.

## NAME

```
describe-instances -
```

## Descrizione

Una descrizione dell'APIoperazione richiamata dal comando.

## 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.

...

## Riepilogo

Sintassi di base per l'utilizzo del comando e delle relative opzioni. Le opzioni tra parentesi quadre sono facoltative, hanno un valore predefinito oppure hanno un'opzione che può essere utilizzata in alternativa.

## 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]
```

Ad esempio, `describe-instances` ha un comportamento predefinito che descrive tutte le istanze nell'account corrente e nella regione. AWS Puoi specificare, a tua discrezione, un elenco di `instance-ids` per descrivere una o più istanze. `dry-run` è un flag booleano opzionale che non utilizza un valore. Per usare un flag booleano specifica il valore mostrato, in questo caso `--dry-run` o `--no-dry-run`. Anche `--generate-cli-skeleton` non utilizza un valore. Se per

l'utilizzo di un'opzione sono previste condizioni, queste sono descritte nella sezione `OPTIONS` o mostrate negli esempi.

## Opzioni

Descrizione di ogni singola opzione mostrata nella sinossi.

### 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.

...

## Esempi

Esempi che mostrano l'utilizzo del comando e delle relative opzioni. Se non è disponibile alcun esempio per un comando o un caso d'uso di cui hai bisogno, richiedine uno utilizzando il link di feedback in questa pagina o nel riferimento al AWS CLI comando nella pagina di aiuto del comando.

### 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

Descrizioni di ciascuno dei campi e dei tipi di dati inclusi nella risposta di AWS.

Per `describe-instances`, l'output è un elenco di oggetti prenotazione, ciascuno dei quali contiene diversi campi e oggetti che includono informazioni sulle istanze a essi associate. Queste informazioni provengono dalla [API documentazione relativa al tipo di dati di prenotazione](#) utilizzato da AmazonEC2.

### 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.

...

```

Quando AWS CLI esegue il rendering dell'output inJSON, questo diventa una matrice di oggetti di prenotazione, simile all'esempio seguente.

```

{
  "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"
          },
        },
      ],
    },
  ],
}
...

```

Ogni oggetto prenotazione contiene campi che descrivono la prenotazione e una serie di oggetti istanza, ognuno con specifici campi (ad esempio, `PublicDnsName`) e oggetti (ad esempio, `State`) che lo descrivono.

### Utenti Windows

Puoi eseguire il piping (`|`) dell'output del comando `help` su `more` per visualizzare il file della guida una pagina alla volta. Premete la barra spaziatrice o `PgDn` per visualizzare altre parti del documento e `q` per uscire.

```
C:\> aws ec2 describe-instances help | more
```

## AWS CLI guida di riferimento

I file della guida contengono collegamenti che non possono essere visualizzati o esplorati dalla riga di comando. È possibile visualizzare e interagire con questi collegamenti utilizzando la guida di [riferimento online AWS CLI versione 1 \(guida di riferimento\)](#). Il riferimento contiene anche il contenuto della guida per tutti AWS CLI i comandi. Le descrizioni sono presentate in modo da semplificare la navigazione e la visualizzazione su dispositivi mobili, tablet o schermi del desktop.

## API documentazione

Tutti i comandi in the AWS CLI corrispondono alle richieste fatte al pubblico di un AWS servizio API. Ogni servizio con un pubblico API ha un API riferimento che può essere trovato nella home page del servizio sul [sito Web della AWS documentazione](#). Il contenuto di un API riferimento varia in base a come API è costruito e al protocollo utilizzato. In genere, un API riferimento contiene informazioni dettagliate sulle operazioni supportate da API, sui dati inviati e dal servizio e sulle eventuali condizioni di errore segnalate dal servizio.

### API Sezioni della documentazione

- **Azioni:** informazioni dettagliate su ciascuna operazione e sui relativi parametri (inclusi vincoli di lunghezza o contenuto e valori predefiniti). Vengono elencati gli errori che possono verificarsi per questa operazione. Ogni operazione corrisponde a un sottocomando in AWS CLI
- **Tipi di dati:** informazioni dettagliate sulle strutture che un comando potrebbe richiedere come parametro o restituire in risposta a una richiesta.
- **Parametri comuni:** informazioni dettagliate sui parametri condivisi da tutte le azioni del servizio.

- Errori comuni: informazioni dettagliate sugli errori che possono essere restituiti da qualsiasi operazione del servizio.

Il nome e la disponibilità di ogni sezione possono variare a seconda del servizio.

### Specifico del servizio CLIs

Alcuni servizi hanno un codice separato CLI che risale a prima che ne AWS CLI venisse creato uno specifico per funzionare con tutti i servizi. Questi servizi specifici CLIs hanno una documentazione separata che è collegata alla pagina della documentazione del servizio. La documentazione relativa ai servizi specifici CLIs non si applica a. AWS CLI

## Risoluzione degli errori

Per informazioni sulla diagnosi e la correzione degli AWS CLI errori, vedere. [Risolvere gli errori](#)

## Aiuto aggiuntivo

Per ulteriore assistenza con i tuoi AWS CLI problemi, visita la [AWS CLI community](#) su GitHub.

## Struttura di comando in AWS CLI

Questo argomento illustra come è strutturato il comando AWS Command Line Interface (AWS CLI) e come utilizzare i comandi wait.

### Argomenti

- [Struttura dei comandi](#)
- [Comandi Wait](#)

## Struttura dei comandi

AWS CLI Utilizza una struttura multiparte sulla riga di comando che deve essere specificata in questo ordine:

1. La chiamata di base al programma aws.
2. Il comando di primo livello, che in genere corrisponde a un AWS servizio supportato da. AWS CLI



3. Il sottocomando che specifica l'operazione da eseguire.
4. AWS CLI Opzioni o parametri generali richiesti dall'operazione. Possono essere specificati in qualsiasi ordine, purché seguano le prime tre parti. Se un parametro esclusivo viene specificato più volte, si applica solo l'ultimo valore.

```
$ aws <command> <subcommand> [options and parameters]
```

I parametri possono assumere vari tipi di valori di input, ad esempio numeri, stringhe, elenchi, mappe e JSON strutture. Quanto supportato dipende dal comando e dal sottocomando specificati.

## Esempi

### Amazon S3

L'esempio seguente elenca tutti i bucket Amazon S3.

```
$ aws s3 ls
2018-12-11 17:08:50 amzn-s3-demo-bucket1
2018-12-14 14:55:44 amzn-s3-demo-bucket2
```

Per ulteriori informazioni sui comandi di Amazon S3, consulta AWS CLI Command [aws s3](#)Reference.

### AWS CloudFormation

Il seguente esempio di [create-change-set](#) comando modifica il nome dello stack di cloudformation in *my-change-set*.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-name my-change-set
```

Per ulteriori informazioni sui AWS CloudFormation comandi, consultate la sezione Command [aws cloudformation](#)Reference.AWS CLI

## Comandi Wait

Alcuni AWS servizi dispongono di `wait` comandi. Qualsiasi comando utilizzato in `aws wait` genere attende il completamento di un comando prima di passare al passaggio successivo. Ciò è particolarmente utile per i comandi multipartite o gli script, poiché è possibile utilizzare un comando `wait` per evitare di passare ai passaggi successivi se il comando `wait` ha esito negativo.

AWS CLI Utilizza una struttura multiparte nella riga di comando per il `wait` comando che deve essere specificata in questo ordine:

1. La chiamata di base al programma `aws`.
2. Il comando di primo livello, che in genere corrisponde a un AWS servizio supportato da AWS CLI
3. Il comando `wait`.
4. Il sottocomando che specifica l'operazione da eseguire.
5. CLI Opzioni o parametri generali richiesti dall'operazione. Possono essere specificati in qualsiasi ordine, purché seguano le prime tre parti. Se un parametro esclusivo viene specificato più volte, si applica solo l'ultimo valore.

```
$ aws <command> wait <subcommand> [options and parameters]
```

I parametri possono assumere vari tipi di valori di input, ad esempio numeri, stringhe, elenchi, mappe e JSON strutture. Quanto supportato dipende dal comando e dal sottocomando specificati.

#### Note

Non tutti i AWS servizi supportano i `wait` comandi. Consulta la [AWS CLI guida](#) di per vedere se il tuo servizio supporta `wait` i comandi.

## Esempi

### AWS CloudFormation

I seguenti esempi di [wait change-set-create-complete](#) comandi si interrompono e riprendono solo dopo aver confermato che `my-change-set` modifica impostata in `my-stack` lo stack è pronto per l'esecuzione.

```
$ aws cloudformation wait change-set-create-complete --stack-name my-stack --change-set-name my-change-set
```

Per ulteriori informazioni sui AWS CloudFormation `wait` comandi, vedere [wait](#) nel AWS CLI Command Reference.

### AWS CodeDeploy

I seguenti esempi di `wait deployment-successful` comandi si interrompono fino al `d-A1B2C3111` la distribuzione viene completata correttamente.

```
$ aws deploy wait deployment-successful --deployment-id d-A1B2C3111
```

Per ulteriori informazioni sui AWS CodeDeploy `wait` comandi, vedere [wait](#) nel AWS CLI Command Reference.

## Specificazione dei valori dei parametri nel AWS CLI

Molti parametri utilizzati in AWS Command Line Interface (AWS CLI) sono semplici stringhe o valori numerici, come il nome della coppia di chiavi `my-key-pair` nell'esempio di comando seguente `aws ec2 create-key-pair`.

```
$ aws ec2 create-key-pair --key-name my-key-pair
```

La formattazione del comando può variare tra i terminali. Ad esempio, la maggior parte dei terminali distingue tra maiuscole e minuscole, ma Powershell non fa distinzione tra maiuscole e minuscole. Ciò significa che i due esempi di comandi seguenti produrrebbero risultati diversi per i terminali con distinzione tra maiuscole e minuscole a seconda della visualizzazione `MyFile*.txt` e `myfile*.txt` dei diversi parametri.

Tuttavia, PowerShell elaborerebbe queste richieste nello stesso modo in cui le vede `MyFile*.txt` e con gli `myfile*.txt` stessi parametri. Il seguente esempio di comando illustra questi parametri utilizzando il comando: `aws s3 cp`

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "MyFile*.txt"
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "myfile*.txt"
```

[Per ulteriori informazioni sulla PowerShell non distinzione tra maiuscole e minuscole, consulta `about\_case-sensitive` nella documentazione. PowerShell](#)

A volte è necessario utilizzare virgolette o caratteri letterali attorno a stringhe che includono caratteri speciali o spaziali. Le regole relative a questa formattazione possono variare anche tra i terminali. Per ulteriori informazioni sull'utilizzo di virgolette attorno a parametri complessi, consulta [Utilizzo di virgolette e valori letterali con stringhe in AWS CLI](#).

Questi argomenti riguardano le regole di formattazione dei terminali più comuni. Se riscontrate problemi con il riconoscimento dei valori dei parametri da parte del terminale, assicuratevi di

consultare gli argomenti di questa sezione e di consultare anche la documentazione del terminale per le relative regole di sintassi specifiche.

## Argomenti sui parametri

- [Tipi di parametri comuni in AWS CLI](#)
- [Utilizzo di virgolette e valori letterali con stringhe in AWS CLI](#)
- [Caricamento dei parametri da un file in AWS CLI](#)
- [AWS CLI scheletri e file di input nel AWS CLI](#)
- [Utilizzo della sintassi abbreviata in AWS CLI](#)

## Tipi di parametri comuni in AWS CLI

Questa sezione descrive alcuni tipi di parametri comuni e il formato tipico richiesto.

Se hai problemi a formattare un parametro per un comando specifico, consulta l'aiuto inserendo **help** dopo il nome del comando. L'aiuto per ogni sottocomando include il nome e la descrizione di un'opzione. Il tipo di parametro dell'opzione è elencato tra parentesi. Per ulteriori informazioni sulla visualizzazione della guida, vedere [the section called “Chiedere aiuto”](#)

I tipi di parametri includono:

- [Stringa](#)
- [Timestamp](#)
- [Elenco](#)
- [Booleano](#)
- [Numero intero](#)
- [Binary/blob \(oggetto binario di grandi dimensioni\) e blob in streaming](#)
- [Eseguire la mappatura](#)
- [Documento](#)

## Stringa

I parametri delle stringhe possono contenere caratteri alfanumerici, simboli e spazi bianchi del set di [ASCII](#) caratteri. Le stringhe che contengono spazi bianchi devono essere racchiuse tra virgolette. Ti consigliamo di non utilizzare simboli o spazi bianchi diversi dal carattere di spazio standard e di osservare le [regole di citazione](#) del tuo terminale per evitare risultati imprevisti.

Alcuni parametri di stringa accettano i dati binari da un file. Consulta [File binari](#) per un esempio.

## Timestamp

[I timestamp sono formattati secondo lo standard 8601. ISO](#) Questi sono spesso denominati parametri "" o DateTime "». Date

```
$ aws ec2 describe-spot-price-history --start-time 2014-10-13T19:00:00Z
```

I formati accettabili includono:

- *YYYY-MM-DDThh:mm:ss.sssTZD (UTC)*, ad esempio, 2014-10-01T 20:30:00.000 Z
- *YYYY-MM-DDThh:mm:ss.sssTZD (with offset)*, ad esempio, 2014-10-01T 12:30:00.000-08:00
- *YYYY-MM-DD*, ad esempio, 2014-10-01
- Tempo Unix in secondi, ad esempio 1412195400. A volte viene chiamata [ora Unix Epoch](#) e rappresenta il numero di secondi trascorsi dalla mezzanotte del 1° gennaio 1970. UTC

È possibile impostare il formato del timestamp utilizzando l'impostazione del file.

[cli\\_timestamp\\_format](#)

## Elenco

Una o più stringhe separate da spazi. Se uno degli elementi della stringa contiene uno spazio, devi racchiuderlo tra virgolette. Osservate le [regole di quotazione](#) del vostro terminale per evitare risultati imprevisti.

```
$ aws ec2 describe-spot-price-history --instance-types m1.xlarge m1.medium
```

## Booleano

Bandiera binaria che attiva o disattiva un'opzione. Ad esempio, `ec2 describe-spot-price-history` ha un parametro `--dry-run` booleano che, quando specificato, consente di convalidare la query con il servizio anche senza eseguire la query.

```
$ aws ec2 describe-spot-price-history --dry-run
```

L'output indica se il formato del comando è valido. Questo comando include anche una versione `--no-dry-run` del parametro, che puoi utilizzare per indicare esplicitamente che il comando deve

essere eseguito normalmente, anche se non è necessario includerlo poiché questa è l'impostazione predefinita.

## Numero intero

Numero intero senza firma.

```
$ aws ec2 describe-spot-price-history --max-items 5
```

## Binary/blob (oggetto binario di grandi dimensioni) e blob in streaming

In AWS CLI, è possibile passare un valore binario come stringa direttamente sulla riga di comando. Esistono due tipi di blob:

- [Blob](#)
- [Blob in streaming](#)

### Blob

Per passare un valore a un parametro con tipo `blob`, è necessario specificare un percorso a un file locale che contiene i dati binari utilizzando il `fileb://` prefisso. I file a cui si fa riferimento utilizzando il `fileb://` prefisso vengono sempre trattati come file binari non codificati non elaborati. Il percorso specificato viene interpretato come relativo alla directory di lavoro. Ad esempio, il parametro `--plaintext` per `aws kms encrypt` è un blob.

```
$ aws kms encrypt \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --plaintext fileb://ExamplePlaintextFile \  
  --output text \  
  --query CiphertextBlob | base64 \  
  --decode > ExampleEncryptedFile
```

### Blob in streaming

I blob di streaming, ad esempio `Do not use aws cloudsearchdomain upload-documents` prefissi. Invece, i parametri dello streaming blob vengono formattati utilizzando il percorso diretto del file. L'esempio seguente utilizza il percorso diretto del file `document-batch.json` per il `aws cloudsearchdomain upload-documents` comando:

```
$ aws cloudsearchdomain upload-documents \  
  --document-batch document-batch.json
```

```
--endpoint-url https://doc-my-domain.us-west-1.cloudsearch.amazonaws.com \  
--content-type application/json \  
--documents document-batch.json
```

## Eseguire la mappatura

Un insieme di coppie chiave-valore specificate in JSON o utilizzando la sintassi [abbreviata CLI](#) di. L'JSONesempio seguente legge un elemento da una tabella Amazon DynamoDB denominata my-table con un parametro map,. --key Il parametro specifica la chiave primaria denominata id con un valore numerico pari a 1 in una struttura annidata. JSON

Per un JSON utilizzo più avanzato in una riga di comando, prendi in considerazione l'utilizzo di un JSON processore da riga di comando, ad esempio jq per creare JSON stringhe. Per maggiori informazioni su jq, consulta il [repository jq](#) su. GitHub

```
$ aws dynamodb get-item --table-name my-table --key '{"id": {"N": "1"}}'  
  
{  
  "Item": {  
    "name": {  
      "S": "John"  
    },  
    "id": {  
      "N": "1"  
    }  
  }  
}
```

## Documento

### Note

La [sintassi abbreviata](#) non è compatibile con i tipi di documento.

I tipi di documento vengono utilizzati per inviare dati senza la necessità di incorporarli all'interno di stringhe. JSON Il tipo di documento consente ai servizi di fornire schemi arbitrari per l'utilizzo di tipi di dati più flessibili.

Ciò consente l'invio di JSON dati senza la necessità di sfuggire ai valori. Ad esempio, invece di utilizzare il seguente input con escapeJSON:

```
{"document": "{\"key\":true}"}
```

È possibile utilizzare il seguente tipo di documento:

```
{"document": {"key": true}}
```

Valori validi per i tipi di documento

A causa della natura flessibile dei tipi di documento, esistono più tipi di valori validi. I valori validi includono i seguenti:

Stringa

```
--option "value"
```

Numero

```
--option 123  
--option 123.456
```

Booleano

```
--option true
```

Null

```
--option null
```

Array

```
--option ["value1", "value2", "value3"]  
--option ["value", 1, true, null, ["key1", 2.34], {"key2": "value2"}]
```

Oggetto

```
--option {"key": "value"}  
--option {"key1": "value1", "key2": 123, "key3": true, "key4": null, "key5":  
["value3", "value4"], "key6": {"value5": "value6"}}
```



## Utilizzo di virgolette e valori letterali con stringhe in AWS CLI

Esistono principalmente due modi in cui vengono utilizzate le virgolette singole e doppie in AWS CLI

- [Utilizzo delle virgolette attorno a stringhe che contengono spazi bianchi](#)
- [Utilizzo delle virgolette all'interno delle stringhe](#)

### Utilizzo delle virgolette attorno a stringhe che contengono spazi bianchi

I nomi dei parametri e i relativi valori sono separati da spazi sulla riga di comando. Se un valore di stringa contiene uno spazio incorporato, è necessario racchiudere l'intera stringa tra virgolette per evitare che lo spazio venga interpretato erroneamente come divisore tra il valore e il nome del parametro successivo. AWS CLI Il tipo di virgolette utilizzato dipende dal sistema operativo su cui è in esecuzione. AWS CLI

#### Linux and macOS

Usa virgolette singole ' '

```
$ aws ec2 create-key-pair --key-name 'my key pair'
```

Per ulteriori informazioni sull'uso delle virgolette, consultate la documentazione per l'utente della vostra shell preferita.

#### PowerShell

Citazioni singole (consigliate)

Le virgolette singole ' ' sono chiamate *verbatim* stringhe. La stringa viene passata al comando esattamente mentre viene digitata, il che significa che PowerShell le variabili non verranno trasmesse.

```
PS C:\> aws ec2 create-key-pair --key-name 'my key pair'
```

#### Virgolette doppie

Le virgolette doppie " " sono chiamate *expandable* stringhe. Le variabili possono essere passate in stringhe espandibili.

```
PS C:\> aws ec2 create-key-pair --key-name "my key pair"
```

Per ulteriori informazioni sull'utilizzo delle virgolette, vedere [Informazioni sulle regole di quotazione](#) in Microsoft PowerShell Docs.

## Windows command prompt

Usa le virgolette doppie. " "

```
C:\> aws ec2 create-key-pair --key-name "my key pair"
```

Facoltativamente, è possibile separare il nome del parametro dal valore con un segno di uguale = anziché uno spazio. In genere è necessario solo se il valore del parametro inizia con un trattino.

```
$ aws ec2 delete-key-pair --key-name=-mykey
```

## Utilizzo delle virgolette all'interno delle stringhe

Le stringhe potrebbero contenere virgolette e la shell potrebbe richiedere le virgolette di escape per farle funzionare correttamente. Uno dei tipi di valori dei parametri più comuni è una stringa. JSON Si tratta di un processo complesso in quanto include spazi e virgolette doppie " " attorno al nome e al valore di ogni elemento nella JSON struttura. Il modo in cui immettete i parametri JSON -formatted nella riga di comando varia a seconda del sistema operativo.

Per un JSON utilizzo più avanzato nella riga di comando, prendi in considerazione l'utilizzo di un JSON processore da riga di comando, ad esempio per creare jq stringhe. JSON Per maggiori informazioni su jq, consulta il [repository jq](#) su. GitHub

## Linux and macOS

Affinché Linux e macOS interpretino le stringhe, usa letteralmente le virgolette singole ' ' per racchiudere la struttura JSON dei dati, come nell'esempio seguente. Non è necessario evitare le virgolette doppie incorporate nella JSON stringa, poiché vengono trattate letteralmente. Poiché JSON è racchiuso tra virgolette singole, sarà necessario eliminare tutte le virgolette singole nella stringa, in genere utilizzando una barra rovesciata prima delle virgolette singole. \'

```
$ aws ec2 run-instances \  
  --image-id ami-12345678 \  
  --block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":  
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}]'
```

Per ulteriori informazioni sull'uso delle virgolette, consultate la documentazione per l'utente della shell preferita.

## PowerShell

Usa virgolette singole ' ' o doppie. " "

### Virgolette singole (consigliate)

Le virgolette singole ' ' sono chiamate `verbatim` stringhe. La stringa viene passata al comando esattamente mentre viene digitata, il che significa che PowerShell le variabili non verranno trasmesse.

Poiché le strutture JSON dati includono virgolette doppie, suggeriamo di racchiuderle tra ' ' virgolette singole. Se si utilizzano virgolette singole, non è necessario evitare le virgolette doppie incorporate nella stringa. JSON Tuttavia, è necessario evitare ogni singola virgoletta inserendo un segno di spunta all'interno della struttura. JSON

```
PS C:\> aws ec2 run-instances `
  --image-id ami-12345678 `
  --block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}]`
```

### Citazioni doppie

Le virgolette doppie " " sono chiamate `expandable` stringhe. Le variabili possono essere passate in stringhe espandibili.

Se si utilizzano le virgolette doppie, non è necessario evitare le virgolette singole incorporate nella stringa. JSON Tuttavia, è necessario evitare ogni virgoletta doppia inserendo un segno di spunta all'interno della JSON struttura, come nell'esempio seguente.

```
PS C:\> aws ec2 run-instances `
  --image-id ami-12345678 `
  --block-device-mappings "[{"DeviceName `": `"/dev/sdb `", `Ebs `":
{ `VolumeSize `":20, `DeleteOnTermination `":false, `VolumeType `": `standard `"}]"
```

Per ulteriori informazioni sull'utilizzo delle virgolette, vedere [Informazioni sulle regole di quotazione](#) in Microsoft PowerShell Docs.

**⚠ Warning**

Prima di PowerShell inviare un comando a AWS CLI, determina se il comando viene interpretato utilizzando regole tipiche PowerShell o di `CommandLineToArgvW` citazione. Quando PowerShell i processi vengono utilizzati `CommandLineToArgvW`, è necessario escludere i caratteri con una barra \ rovesciata.

Per ulteriori informazioni su *CommandLineToArgvW* in PowerShell, vedi [What's up with the strange treatment of virgolette and backslashes di CommandLineToArgvW](#) in Microsoft DevBlogs, [Everyone cita gli argomenti della riga di comando nel modo sbagliato](#) nel Microsoft Docs Blog e la funzione [CommandLineToArgvW](#) in Microsoft Docs.

**Citazioni singole**

Le virgolette singole ' ' sono chiamate `verbatim` stringhe. La stringa viene passata al comando esattamente mentre viene digitata, il che significa che PowerShell le variabili non verranno trasmesse. Fuggi dai personaggi con una barra rovesciata \.

```
PS C:\> aws ec2 run-instances `
  --image-id ami-12345678 `
  --block-device-mappings '[{"DeviceName\":"\\/dev/sdb\\",\"Ebs\":"
  {\"VolumeSize\":20,\"DeleteOnTermination\":false,\"VolumeType\":"standard\"}]`
```

**Citazioni doppie**

Le virgolette doppie " " sono chiamate `expandable` stringhe. Le variabili possono essere passate sotto forma di `stringheexpandable`. Per le stringhe tra virgolette devi scappare due volte usando ` ` per ogni citazione invece di usare solo un backtick. Il backtick sfugge alla barra rovesciata, quindi la barra rovesciata viene utilizzata come carattere di escape per il processo. `CommandLineToArgvW`

```
PS C:\> aws ec2 run-instances `
  --image-id ami-12345678 `
  --block-device-mappings "[{\"DeviceName `\": `\\/dev/sdb `\", `\"Ebs `\":
  { `\"VolumeSize `\":20, `\"DeleteOnTermination `\":false, `\"VolumeType `\": `
  `\"standard `\"}]`"
```

**Blobs (consigliato)**

Per aggirare le regole PowerShell di citazione per l'immissione JSON dei dati, usa `Blobs` per passare i dati direttamente a. `JSON AWS CLI` Per ulteriori informazioni su `Blobs`, vedere. [Blob](#)

## Windows command prompt

Il prompt dei comandi di Windows richiede virgolette doppie " " per racchiudere la struttura dei dati. JSON Inoltre, per evitare che il processore di comandi interpreti erroneamente le virgolette doppie incorporate " in JSON \ JSON

```
C:\> aws ec2 run-instances ^
  --image-id ami-12345678 ^
  --block-device-mappings "[{\\"DeviceName\\":\\"/dev/sdb\\",\\"Ebs\\":
{\\"VolumeSize\\":20,\\"DeleteOnTermination\\":false,\\"VolumeType\\":\\"standard\\"}]"
```

Soltanto le virgolette doppie più esterne non richiedono il carattere di escape.

## Caricamento dei parametri da un file in AWS CLI

Alcuni parametri prevedono i nomi dei file come argomenti, dai quali vengono AWS CLI caricati i dati. Altri parametri consentono di specificare il valore del parametro come testo digitato sulla riga di comando o letto da un file. Indipendentemente dal fatto che un file sia obbligatorio o facoltativo, è necessario codificarlo correttamente in modo che AWS CLI possano comprenderlo. La codifica del file deve corrispondere alle impostazioni locali predefinite del sistema di lettura. È possibile determinarlo utilizzando il metodo `Pythonlocale.getpreferredencoding()`.

### Note

Per impostazione predefinita, Windows PowerShell restituisce il testo come UTF -16, il che è in conflitto con la codifica UTF -8 utilizzata dai JSON file e da molti sistemi Linux. Si consiglia di utilizzarlo `-Encoding ascii` con i PowerShell `Out-File` comandi per assicurarsi che siano in AWS CLI grado di leggere il file risultante.

## Argomenti

- [Come caricare i parametri da un file](#)
- [File binari](#)
- [File remoti](#)

## Come caricare i parametri da un file

A volte è conveniente caricare il valore di un parametro da un file anziché provare a digitarlo tutto come valore di parametro della riga di comando, ad esempio quando il parametro è una JSON stringa complessa. Per specificare un file che contiene il valore, specificate un file URL nel formato seguente.

```
file://complete/path/to/file
```

- I primi due caratteri di barra "/" fanno parte della specifica. Se il percorso richiesto inizia con "/", il risultato conterrà tre caratteri di barra: `file:///folder/file`.
- URL fornisce il percorso del file che contiene il contenuto effettivo dei parametri.
- Quando utilizzate file con spazi o caratteri speciali, seguite le [regole di virgolette ed escape](#) previste dal vostro terminale.

### Note

Questo comportamento è disabilitato automaticamente per i parametri che già prevedono un URL, ad esempio il parametro che identifica un modello. AWS CloudFormation URL Puoi anche disabilitare questo comportamento disabilitando l'[cli\\_follow\\_urlparam](#) impostazione nel tuo AWS CLI file di configurazione.

I percorsi di file specificati nei seguenti esempi vengono interpretati come relativi alla directory di lavoro corrente.

### 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"
```

L'opzione del prefisso `file://` supporta le espansioni in stile Unix, tra cui `~/`, `./` e `../`. In Windows l'espressione `~/` si espande alla directory utente, archiviata nella variabile di ambiente `%USERPROFILE%`. Ad esempio, in Windows 10 in genere è presente una directory utente in `C:\Users\UserName\`.

È comunque necessario evitare JSON i documenti incorporati come valore di un altro JSON documento.

```
$ 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\\"}"
}
```

## File binari

Per i comandi che richiedono dati binari come parametro, è necessario specificare che i dati sono contenuto binario mediante il prefisso `fileb://`. I comandi che accettano i dati binari includono:

- **aws ec2 run-instances:** `--user-dataparametro`.
- **aws s3api put-object:** `--sse-customer-keyparametro`.
- **aws kms decrypt:** `--ciphertext-blobparametro`.

L'esempio seguente genera una AES chiave binaria a 256 bit utilizzando uno strumento da riga di comando Linux, quindi la fornisce ad Amazon S3 per crittografare un file caricato lato server.

```
$ 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\""
}

```

## File remoti

Supporta AWS CLI anche il caricamento dei parametri da un file ospitato su Internet con un `or`. `http://` `https://` URL L'esempio seguente fa riferimento a un file archiviato in un bucket Amazon S3. Questo consente di accedere ai file di parametri da qualsiasi computer, ma richiede che il container sia accessibile pubblicamente.

```

$ aws ec2 run-instances \
  --image-id ami-12345678 \
  --block-device-mappings http://amzn-s3-demo-bucket.s3.amazonaws.com/filename.json

```

L'esempio precedente presuppone che il file `filename.json` contenga i seguenti dati. JSON

```

[
  {
    "DeviceName": "/dev/sdb",
    "Ebs": {
      "VolumeSize": 20,
      "DeleteOnTermination": false,
      "VolumeType": "standard"
    }
  }
]

```

Per un altro esempio di riferimento a un file contenente parametri JSON `-formatted`, vedere. [Allegare una politica IAM gestita a un utente](#)



## AWS CLI scheletri e file di input nel AWS CLI

La maggior parte dei AWS CLI comandi accetta tutti gli input di parametri da un file. Questi modelli possono essere generati utilizzando l'`generate-cli-skeleton` opzione.

### Argomenti

- [Informazioni sugli AWS CLI scheletri e sui file di input](#)
- [Generazione di uno scheletro di comandi](#)

### Informazioni sugli AWS CLI scheletri e sui file di input

La maggior parte dei comandi AWS Command Line Interface (AWS CLI) supporta la possibilità di accettare tutti gli input di parametri da un file utilizzando il `--cli-input-yaml` parametro `--cli-input-json`.

Questi stessi comandi forniscono utilmente il `--generate-cli-skeleton` parametro per generare un file in un YAML formato con tutti i parametri che è possibile modificare e compilare. Quindi puoi eseguire il comando con il parametro `--cli-input-json` appropriato e scegliere il file compilato.

#### Important

Diversi AWS CLI comandi non sono mappati direttamente alle singole AWS API operazioni, come i [aws s3](#). Questi comandi non supportano i parametri `--generate-cli-skeleton`, `--cli-input-json` trattati in questo argomento. Se non sai se un comando specifico supporta questi parametri, esegui il comando seguente, sostituendo il *service* e *command* nomi con quelli che ti interessano.

```
$ aws service command help
```

L'output include una sezione *Synopsis* che mostra i parametri supportati dal comando specificato.

```
$ aws iam list-users help
...
SYNOPSIS
    list-users
    ...
    [--cli-input-json]
    ...
```

```
[--generate-cli-skeleton <value>]
```

```
...
```

Il parametro `--generate-cli-skeleton` impedisce l'esecuzione del comando, generando e visualizzando invece un modello di parametri personalizzabile che può essere utilizzato come input in un secondo momento. Il modello generato include tutti i parametri supportati dal comando.

Il parametro `--generate-cli-skeleton` accetta uno dei seguenti valori:

- `input`— Il modello generato include tutti i parametri di input formattati come JSON. Si tratta del valore di default.
- `output`— Il modello generato include tutti i parametri di output formattati come JSON.

Poiché AWS CLI è essenzialmente un «involucro» attorno al servizio API, il file skeleton si aspetta che tu faccia riferimento a tutti i parametri tramite i nomi dei parametri sottostanti. API Probabilmente è diverso dal nome del parametro. AWS CLI Ad esempio, un AWS CLI parametro denominato `username` potrebbe essere mappato al API parametro denominato del AWS servizio `UserName` (notate le maiuscole alterate e il trattino mancante). Ti consigliamo di utilizzare l'opzione `--generate-cli-skeleton` per generare il modello con i nomi di parametro "corretti" per evitare errori. Puoi anche fare riferimento alla Guida API di riferimento per il servizio per vedere i nomi dei parametri previsti. Puoi eliminare dal modello tutti i parametri che non sono obbligatori e per i quali non desideri fornire un valore.

Ad esempio, se esegui il comando seguente, viene generato il modello di parametro per il comando `run-instances` Amazon Elastic Compute Cloud (AmazonEC2).

## JSON

L'esempio seguente mostra come generare un modello formattato in JSON utilizzando il valore predefinito (`input`) per il `--generate-cli-skeleton` parametro.

```
$ 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
}
```

## Generazione di uno scheletro di comandi

Per generare e utilizzare un file scheletro dei parametri

1. Esegui il comando con il `--generate-cli-skeleton` parametro per produrre e indirizza l'output su un file per salvarlo.

JSON

```
$ aws ec2 run-instances --generate-cli-skeleton input > ec2runinst.json
```

2. Apri il file scheletro dei parametri nell'editor di testo e rimuovi qualsiasi parametro non necessario. Ad esempio, è possibile eliminare il modello fino al seguente. Assicuratevi che il file sia ancora valido JSON dopo aver rimosso gli elementi non necessari.

## JSON

```
{
  "DryRun": true,
  "ImageId": "",
  "KeyName": "",
  "SecurityGroups": [
    ""
  ],
  "InstanceType": "",
  "Monitoring": {
    "Enabled": true
  }
}
```

In questo esempio, lasciamo il `DryRun` parametro impostato `true` per utilizzare la funzionalità Amazon EC2 dry run. Questa funzione consente di testare in modo sicuro il comando senza creare o modificare effettivamente alcuna risorsa.

3. Compila i valori rimanenti con valori appropriati per il tuo scenario. In questo esempio, forniamo il tipo di istanza, il nome della chiave, il gruppo di sicurezza e l'identificatore di Amazon Machine Image (AMI) da utilizzare. Questo esempio presuppone la regione predefinita AWS . AMI `ami-dfc39aef` è un'immagine Amazon Linux a 64 bit ospitata nella `us-west-2` regione. Se utilizzi una regione diversa, devi [trovare l'AMIID corretto da utilizzare](#).

## JSON

```
{
  "DryRun": true,
  "ImageId": "ami-dfc39aef",
  "KeyName": "mykey",
  "SecurityGroups": [
    "my-sg"
  ],
  "InstanceType": "t2.micro",
  "Monitoring": {
    "Enabled": true
  }
}
```

4. AWS CLI interpreta il percorso in modo che sia relativo alla directory di lavoro corrente, quindi nell'esempio seguente, che visualizza solo il nome del file senza percorso, cerca il file direttamente nella directory di lavoro corrente.

## 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.
```

L'errore dry run indica che JSON è formato correttamente e che i valori del parametro sono validi. Se nell'output vengono segnalati altri problemi, correggerli e ripetere la fase precedente fino a quando non viene visualizzato il messaggio "Request would have succeeded".

5. Ora puoi impostare il parametro DryRun su `false` per disabilitare il test.

## JSON

```
{
  "DryRun": false,
  "ImageId": "ami-dfc39aef",
  "KeyName": "mykey",
  "SecurityGroups": [
    "my-sg"
  ],
  "InstanceType": "t2.micro",
  "Monitoring": {
    "Enabled": true
  }
}
```

6. Esegui il comando e avvia `run-instances` effettivamente un'EC2istanza Amazon e visualizza i dettagli generati dal lancio riuscito. Il formato dell'output è controllato dal parametro `--output`, separatamente dal formato del modello di parametro di input.

## JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json --output json
```

```
{
  "OwnerId": "123456789012",
  "ReservationId": "r-d94a2b1",
  "Groups": [],
  "Instances": [
    ...
  ]
}
```

## Utilizzo della sintassi abbreviata in AWS CLI

Il AWS Command Line Interface (AWS CLI) può accettare molti dei suoi parametri di opzione in formato JSON. Tuttavia, può essere noioso inserire JSON elenchi o strutture di grandi dimensioni sulla riga di comando. Per semplificare questa operazione, supporta AWS CLI anche una sintassi abbreviata che consente una rappresentazione più semplice dei parametri delle opzioni rispetto all'utilizzo del formato completo JSON.

### Argomenti

- [Parametri della struttura](#)
- [Utilizzo della sintassi abbreviata con AWS Command Line Interface](#)

### Parametri della struttura

La sintassi abbreviata di AWS CLI semplifica l'immissione di parametri piatti (strutture non annidate) da parte degli utenti. Il formato è un elenco separato da virgole di coppie chiave-valore. Assicuratevi di utilizzare le regole di [quotazione](#) ed escape appropriate per il vostro terminale, poiché la sintassi abbreviata è costituita da stringhe.

### Linux or macOS

```
--option key1=value1,key2=value2,key3=value3
```

### PowerShell

```
--option "key1=value1,key2=value2,key3=value3"
```

Sono entrambe equivalenti al seguente esempio, formattato in JSON.

```
--option '{"key1":"value1","key2":"value2","key3":"value3"}
```

Le coppie chiave-valore separate da virgole non devono contenere spazi vuoti. Ecco un esempio del comando Amazon DynamoDB con l'opzione `update-table` specificata in forma `--provisioned-throughput` abbreviata.

```
$ aws dynamodb update-table \
  --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10 \
  --table-name MyDDBTable
```

Questo è equivalente al seguente esempio formattato in JSON

```
$ aws dynamodb update-table \
  --provisioned-throughput '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}' \
  --table-name MyDDBTable
```

## Utilizzo della sintassi abbreviata con AWS Command Line Interface

È possibile specificare i parametri di input sotto forma di elenco in due modi: o in forma abbreviata. JSON La sintassi abbreviata di AWS CLI è stata progettata per rendere più facile il trasferimento di elenchi di numeri, di stringhe o di strutture non nidificate.

Qui viene mostrato il formato di base, in cui i valori nell'elenco sono separati da un solo spazio.

```
--option value1 value2 value3
```

È equivalente al seguente esempio, formattato in JSON

```
--option '[value1,value2,value3]'
```

Come accennato in precedenza, è possibile specificare un elenco di numeri, un elenco di stringhe o un elenco di strutture non nidificate in formato abbreviato. Di seguito è riportato un esempio del `stop-instances` comando per Amazon Elastic Compute Cloud (AmazonEC2), in cui il parametro di input (elenco di stringhe) per l'`--instance-ids` opzione è specificato in forma abbreviata.

```
$ aws ec2 stop-instances \
  --instance-ids i-1486157a i-1286157c i-ec3a7e87
```



Questo è equivalente al seguente esempio formattato in JSON

```
$ aws ec2 stop-instances \  
  --instance-ids '["i-1486157a","i-1286157c","i-ec3a7e87"]'
```

L'esempio seguente mostra il EC2 create-tags comando Amazon, che richiede un elenco di strutture non annidate per l'--tagsopzione. L'opzione --resources specifica l'ID dell'istanza da contrassegnare.

```
$ aws ec2 create-tags \  
  --resources i-1286157c \  
  --tags Key=My1stTag,Value=Value1 Key=My2ndTag,Value=Value2  
  Key=My3rdTag,Value=Value3
```

Questo è equivalente al seguente esempio, formattato in JSON Il JSON parametro è scritto su più righe per garantire la leggibilità.

```
$ aws ec2 create-tags \  
  --resources i-1286157c \  
  --tags '['  
    {"Key": "My1stTag", "Value": "Value1"},  
    {"Key": "My2ndTag", "Value": "Value2"},  
    {"Key": "My3rdTag", "Value": "Value3"}  
  ]'
```

## Controllo dell'output dei comandi in AWS CLI

Questa sezione descrive i diversi modi per controllare l'output di AWS Command Line Interface (AWS CLI). La personalizzazione dell' AWS CLI output nel terminale può migliorare la leggibilità, semplificare l'automazione degli script e facilitare la navigazione tra set di dati più grandi.

AWS CLI Alcuni servizi prevedono l'[impaginazione](#) lato server per i dati impaginazione aggiuntive.

Infine, AWS CLI dispone di filtri sia lato [server che lato client che è possibile utilizzare singolarmente o insieme per filtrare l'output](#). AWS CLI

### Argomenti

- [Uscita sensibile](#)
- [Opzioni di output lato server o lato client](#)

- [Impostazione del formato di output in AWS CLI](#)
- [Utilizzo delle opzioni di impaginazione in AWS CLI](#)
- [Uscita di filtraggio nel AWS CLI](#)

## Uscita sensibile

Alcune operazioni di AWS CLI potrebbero restituire informazioni che potrebbero essere considerate riservate, incluse le informazioni provenienti da variabili di ambiente. L'esposizione di queste informazioni potrebbe rappresentare un rischio per la sicurezza in determinati scenari; ad esempio, le informazioni potrebbero essere incluse nei registri di integrazione e distribuzione continua (CI/CD). È quindi importante verificare quando includere tale output nei log e sopprimerlo quando non è necessario.

Per ulteriori informazioni sulla protezione dei dati sensibili, vedere. [the section called “Protezione dei dati”](#)

Considerate le seguenti best practice:

- Prendi in considerazione la possibilità di recuperare a livello di codice i tuoi segreti da un archivio di segreti, ad esempio. AWS Secrets Manager
- Esamina il contenuto dei registri di compilazione per assicurarti che non contengano informazioni riservate. Prendi in considerazione approcci come il piping `/dev/null` o l'acquisizione dell'output come bash o PowerShell variabile per sopprimere gli output dei comandi.

Quello che segue è un esempio di bash per reindirizzare l'output, ma non gli errori, a: `/dev/null`

```
$ aws s3 ls > /dev/null
```

Per informazioni specifiche sulla soppressione dell'output per il terminale, consultate la documentazione per l'utente del terminale che utilizzate.

- Prendi in considerazione l'accesso ai tuoi log e definisci l'accesso in modo appropriato per il tuo caso d'uso.

## Opzioni di output lato server o lato client

AWS CLI Dispone di filtri sia lato [server che lato client che puoi utilizzare singolarmente o insieme per filtrare l'output](#). AWS CLI Il filtraggio lato server viene elaborato per primo e restituisce l'output

per il filtraggio lato client. Il filtraggio lato server è supportato dal servizio. API Il filtraggio lato client è supportato dal client che utilizza il parametro. AWS CLI `--query`

Le opzioni di output lato server sono funzionalità supportate direttamente da. Servizio AWS API Tutti i dati filtrati o eliminati non vengono inviati al client, il che può accelerare i tempi di HTTP risposta e migliorare la larghezza di banda per set di dati più grandi.

Le opzioni di output lato client sono funzionalità create da. AWS CLI Tutti i dati vengono inviati al client, quindi i AWS CLI filtri o le pagine in cui vengono visualizzati i contenuti. Le operazioni lato client non consentono di risparmiare sulla velocità o sulla larghezza di banda per set di dati più grandi.

Quando le opzioni lato server e lato client vengono utilizzate insieme, le operazioni lato server vengono prima completate e poi inviate al client per le operazioni lato client. Ciò sfrutta i potenziali risparmi di velocità e larghezza di banda delle opzioni lato server, utilizzando al contempo funzionalità aggiuntive per ottenere l'output desiderato. AWS CLI

## Impostazione del formato di output in AWS CLI

Questo argomento descrive i diversi formati di output per AWS Command Line Interface (AWS CLI). AWS CLI Supporta i seguenti formati di output:

- **json**— L'output è formattato come [JSON](#)stringa.
- **text**— L'output è formattato come più righe di valori di stringa separati da tabulazioni. Questo può essere utile per passare l'output a un elaboratore di testi, ad esempio `grep`, `sed` o `awk`.
- **table**— L'output viene formattato come tabella utilizzando i caratteri `+`/`-` per formare i bordi delle celle. In genere presenta le informazioni in un formato comprensibile molto più semplice da leggere rispetto ad altri, ma non altrettanto utile a livello programmatico.

### Come scegliere il formato di output

Come illustrato nell'argomento relativo alla [configurazione](#), puoi specificare il formato di output in tre modi diversi:

- Utilizzo dell'**output**opzione in un profilo denominato nel **config** file: l'esempio seguente imposta il formato di output predefinito su. `text`

```
[default]
output=text
```

- Utilizzo della variabile di **AWS\_DEFAULT\_OUTPUT** ambiente: l'output seguente imposta il formato dei comandi in questa sessione della riga di comando fino alla modifica della variabile o al termine della sessione. `table` L'utilizzo di questa variabile di ambiente sostituisce il valore impostato nel file `config`.

```
$ export AWS_DEFAULT_OUTPUT="table"
```

- Utilizzo dell'opzione **--output** sulla riga di comando: l'esempio seguente imposta l'output di questo solo comando su `json`. L'utilizzo di questa opzione con il comando sostituisce eventuali variabili di ambiente attualmente impostate o il valore nel file `config`.

```
$ aws swf list-domains --registration-status REGISTERED --output json
```

### Important

Il tipo di output specificato modifica il funzionamento dell'opzione `--query`:

- Se si specifica `--output text`, l'output viene impaginato prima dell'applicazione del `--query` filtro e quindi AWS CLI esegue la query una volta su ogni pagina dell'output. Per questo motivo, la query include il primo elemento corrispondente in ogni pagina, il che può generare un output aggiuntivo inaspettato. Per filtrare ulteriormente l'output, puoi utilizzare altri strumenti da riga di comando come `head` o `tail`.
- Se si specifica `--output json` l'output viene completamente elaborato come un'unica struttura nativa prima dell'applicazione del `--query` filtro. AWS CLI esegue la query solo una volta sull'intera struttura, producendo un risultato filtrato che viene quindi emesso.

## JSONformato di output

[JSON](#) è il formato di output predefinito di AWS CLI. La maggior parte dei linguaggi di programmazione è in grado di JSON decodificare facilmente le stringhe utilizzando funzioni integrate o librerie disponibili pubblicamente. È possibile combinare l'JSONoutput con l'[opzione --query](#) in modi potenti per filtrare e formattare l' AWS CLI JSONoutput in formato -.

Per un filtraggio più avanzato che potresti non essere in grado di utilizzare `--query`, puoi prendere in considerazione un processore a riga di `jq` comando. JSON Il tutorial ufficiale è disponibile per il download all'indirizzo <http://stedolan.github.io/jq/>.

Di seguito è riportato un esempio di JSON output.

```
$ 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"
    }
  ]
}
```

## Formato di output di testo

Il `text` formato organizza l' AWS CLI output in righe delimitate da tabulazioni. Funziona bene con gli strumenti di testo Unix tradizionali come `grep`, e `sedawk`, e con l'elaborazione del testo eseguita da PowerShell

Il formato di output `text` segue la struttura di base riportata di seguito. Le colonne sono ordinate alfabeticamente in base ai nomi chiave corrispondenti dell'oggetto sottostante. JSON

```
IDENTIFIER sorted-column1 sorted-column2
```

```
IDENTIFIER2 sorted-column1 sorted-column2
```

Segue un esempio di output `text`. Ogni campo è una scheda separata dalle altre, con una scheda aggiuntiva dove c'è un campo vuoto.

```
$ 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
```

La quarta colonna è il campo `PasswordLastUsed` ed è vuota per le ultime due voci perché gli utenti non accedono mai alla AWS Management Console.

### Important

Se specifichi l'output `text`, ti consigliamo di utilizzare sempre l'opzione `--query` per garantire un comportamento coerente.

Questo perché il formato di testo ordina alfabeticamente le colonne di output in base al nome chiave dell'JSON oggetto sottostante restituito dal AWS servizio e risorse simili potrebbero non avere gli stessi nomi di chiave. Ad esempio, la JSON rappresentazione di un'istanza EC2 Amazon basata su Linux potrebbe contenere elementi che non sono presenti nella JSON rappresentazione di un'istanza basata su Windows o viceversa. Inoltre, negli aggiornamenti futuri delle risorse potrebbero essere aggiunti o eliminati elementi chiave-valore che modificano l'ordinamento della colonna. In questa situazione `--query` potenzia la funzionalità dell'output di testo `text` per consentire il controllo completo sul formato dell'output.

Nell'esempio seguente, il comando specifica gli elementi da visualizzare e definisce l'ordinamento delle colonne con la notazione elenco `[key1, key2, ...]`. In questo modo gli utenti hanno la certezza che i valori chiave siano visualizzati sempre nella colonna prevista. Infine, notate come gli AWS CLI output siano il valore di `None` chiavi che non esistono.

```
$ 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
```

L'esempio seguente mostra in che modo utilizzare `grep` e `awk` con l'output text dal comando `aws ec2 describe-instances`. Il primo comando visualizza la zona di disponibilità, lo stato corrente e l'ID istanza di ogni istanza nell'output text. Il secondo comando elabora l'output per visualizzare solo l'istanza IDs di tutte le istanze in esecuzione nella zona di `us-west-2a` disponibilità.

```
$ 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
```

L'esempio seguente fa un passo in più e mostra non solo come filtrare l'output, ma anche come utilizzarlo per automatizzare la modifica dei tipi di istanza per ogni istanza arrestata.

```
$ 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
```

L'output può essere utile anche in PowerShell. Poiché le colonne in text output sono delimitate da tabulazioni, puoi facilmente dividere l'output in un array utilizzando il PowerShell `t` delimitatore. Il comando seguente mostra il valore della terza colonna (InstanceId) se la prima colonna (AvailabilityZone) corrisponde alla stringa us-west-2a.

```
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
```

Nota che, sebbene l'esempio precedente mostri come utilizzare il `--query` parametro per analizzare JSON gli oggetti sottostanti ed estrarre la colonna desiderata, PowerShell ha una sua capacità di gestione JSON, se la compatibilità multiplatforma non è un problema. Invece di gestire l'output come testo, come richiede la maggior parte delle shell di comando, PowerShell consente di utilizzare il `ConvertFrom-JSON` cmdlet per produrre un oggetto strutturato gerarchicamente. È quindi possibile accedere direttamente al membro desiderato da tale oggetto.

```
(aws ec2 describe-instances --output json | ConvertFrom-
Json).Reservations.Instances.InstanceId
```

### Tip

Se l'output è di testo e si filtra l'output in un unico campo utilizzando il parametro `--query`, l'output è una singola riga di valori separati da tabulazione. Per ottenere ogni valore su una riga separata, puoi inserire il campo di output tra parentesi, come mostrato negli esempi seguenti:

Output a riga singola, separato da tabulazione:

```
$ aws iam list-groups-for-user --user-name susan --output text --query
"Groups[].GroupName"
```



```
HRDepartment    Developers    SpreadsheetUsers    LocalAdmins
```

Ogni valore nella propria riga inserendo [GroupName] tra parentesi:

```
$ aws iam list-groups-for-user --user-name susan --output text --query
"Groups[].[GroupName]"
```

```
HRDepartment
Developers
SpreadsheetUsers
LocalAdmins
```

## Formato di output di tabella

Il formato `table` produce rappresentazioni leggibili degli output complessi di AWS CLI in formato tabulare.

```
$ 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          ||
```

```
+-----+
+
```

Puoi combinare l'opzione `--query` con il formato `table` per visualizzare un set di elementi preselezionati dall'output non elaborato. Osserva le differenze di output nelle notazioni dizionario ed elenco: nel primo esempio, i nomi delle colonne sono in ordine alfabetico, mentre nel secondo esempio le colonne senza nome sono ordinate in base a quanto definito dall'utente. Per ulteriori informazioni sull'opzione `--query`, consulta [Uscita di filtraggio nel 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  |
+-----+-----+-----+-----+
```

## Utilizzo delle opzioni di impaginazione in AWS CLI

Questo argomento descrive i diversi modi per impaginare l'output di AWS CLI

## Paginazione lato server

Per i comandi che possono restituire un ampio elenco di elementi, the AWS Command Line Interface (AWS CLI) ha diverse opzioni per controllare il numero di elementi inclusi nell'output quando viene AWS CLI chiamato un servizio API per compilare l'elenco.

Le opzioni sono le seguenti:

- [Come usare il parametro `--no-paginate`](#)
- [Come usare il parametro `--page-size`](#)
- [Come usare il parametro `--max-items`](#)
- [Come usare il parametro `--starting-token`](#)

Per impostazione predefinita, AWS CLI utilizza una dimensione di pagina determinata dal singolo servizio e recupera tutti gli elementi disponibili. Ad esempio, Amazon S3 ha una dimensione di pagina predefinita di 1000. Se esegui `aws s3api list-objects` su un bucket Amazon S3 che contiene 3.500 oggetti, effettua AWS CLI automaticamente quattro chiamate ad Amazon S3, gestendo la logica di impaginazione specifica del servizio in background e restituendo tutti i 3.500 oggetti nell'output finale.

Come usare il parametro `--no-paginate`

L'`--no-paginate` opzione disabilita i seguenti token di impaginazione sul lato client. Quando si utilizza un comando, per impostazione predefinita effettua AWS CLI automaticamente più chiamate per restituire tutti i risultati possibili per creare l'impaginazione. Una chiamata per ogni pagina. La disattivazione della paginazione comporta l' AWS CLI unica chiamata per la prima pagina dei risultati del comando.

Ad esempio, se esegui `aws s3api list-objects` su un bucket Amazon S3 che contiene 3.500 oggetti, effettua AWS CLI solo la prima chiamata ad Amazon S3, restituendo solo i primi 1.000 oggetti nell'output finale.

```
$ aws s3api list-objects \  
  --bucket amzn-s3-demo-bucket \  
  --no-paginate  
{  
  "Contents": [  
  ...
```

## Come usare il parametro `--page-size`

Se riscontri problemi durante l'esecuzione dei comandi di elenco su un gran numero di risorse, la dimensione predefinita della pagina potrebbe essere troppo alta. Ciò può far sì che le chiamate ai AWS servizi superino il tempo massimo consentito e generare un errore di «timeout». È possibile utilizzare l'opzione `--page-size` per specificare che si AWS CLI richiede un numero inferiore di elementi per ogni chiamata al AWS servizio. Continua AWS CLI a recuperare l'elenco completo, ma esegue un numero maggiore di API chiamate di servizio in background e recupera un numero inferiore di elementi a ogni chiamata. In questo modo, le chiamate individuali hanno una migliore possibilità di completamento prima della scadenza. La modifica delle dimensioni della pagina non influisce sull'output, ma solo sul numero di API chiamate da effettuare per generare l'output.

```
$ aws s3api list-objects \  
  --bucket amzn-s3-demo-bucket \  
  --page-size 100  
{  
  "Contents": [  
  ...
```

## Come usare il parametro `--max-items`

Per includere meno elementi alla volta nell' AWS CLI output, usa l'opzione `--max-items`. The AWS CLI still gestisce l'impaginazione con il servizio come descritto in precedenza, ma stampa solo il numero di elementi alla volta specificato.

```
$ aws s3api list-objects \  
  --bucket amzn-s3-demo-bucket \  
  --max-items 100  
{  
  "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxZjQ==",  
  "Contents": [  
  ...
```

## Come usare il parametro `--starting-token`

Se il numero di elementi output (`--max-items`) è inferiore al numero totale di elementi restituiti dalle API chiamate sottostanti, l'output include un elemento `NextToken` che è possibile passare a un comando successivo per recuperare il set di elementi successivo. L'esempio seguente mostra come usare il valore `NextToken` restituito dall'esempio precedente e consente di recuperare il secondo centinaio di elementi.

**Note**

Il parametro `--starting-token` non può essere null o vuoto. Se il comando precedente non restituisce un valore `NextToken`, non esistono altre voci da restituire e non dovrai richiamare di nuovo il comando.

```
$ aws s3api list-objects \  
  --bucket amzn-s3-demo-bucket \  
  --max-items 100 \  
  --starting-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ==\  
{  
  "Contents": [  
  ...
```

Il AWS servizio specificato potrebbe non restituire gli elementi nello stesso ordine ogni volta che si chiama. Se si specificano valori diversi per `--page-size` e `--max-items`, puoi ottenere risultati imprevisti con elementi mancanti o duplicati. Per evitarlo, utilizza lo stesso numero per `--page-size` e `--max-items` per sincronizzare la paginazione della AWS CLI con quella del servizio sottostante. Puoi anche recuperare l'intero elenco ed eseguire tutte le operazioni di paginazione necessarie in locale.

## Uscita di filtraggio nel AWS CLI

AWS Command Line Interface (AWS CLI) dispone di filtri sia sul lato server che sul lato client che è possibile utilizzare singolarmente o insieme per filtrare l'output. AWS CLI Il filtraggio lato server viene elaborato per primo e restituisce l'output per il filtraggio lato client.

- Il filtraggio lato server è supportato da, e in genere viene implementato con un parametro API. `--filter` Il servizio restituisce solo risultati corrispondenti che possono accelerare i tempi di HTTP risposta per set di dati di grandi dimensioni.
- Il filtraggio lato client è supportato dal AWS CLI client che utilizza il parametro. `--query` Questo parametro ha funzionalità che il filtraggio lato server potrebbe non avere.

### Argomenti

- [Filtraggio lato server](#)
- [Filtraggio lato client](#)

- [Combinazione del filtraggio lato server e lato client](#)
- [Risorse aggiuntive](#)

## Filtraggio lato server

Il filtraggio lato server in è fornito dal AWS CLI servizio. AWS API Il AWS servizio restituisce solo i record nella HTTP risposta che corrispondono al filtro, il che può accelerare i tempi di HTTP risposta per set di dati di grandi dimensioni. Poiché il filtraggio lato server è definito dal servizioAPI, i nomi e le funzioni dei parametri variano tra i servizi. Alcuni nomi di parametri comuni utilizzati per il filtraggio sono:

- `--filter` come [ses](#) e [ce ce](#)
- `--filters` <https://docs.aws.amazon.com/cli/latest/reference/rds/describe-db-instances.html>
- Nomi che iniziano con la parola, ad esempio per il comando. `filter --filter-expression aws dynamodb scan`

Per informazioni sull'eventuale presenza di filtri sul lato server e sulle regole di filtraggio di uno specifico comando, consultate la guida di riferimento 2 della [guida AWS CLI di riferimento](#).

## Filtraggio lato client

AWS CLI Fornisce funzionalità di filtraggio lato client integrate JSON con il parametro. `--query` Il `--query` parametro è uno strumento potente che puoi utilizzare per personalizzare il contenuto e lo stile del tuo output. Il `--query` parametro prende la HTTP risposta che arriva dal server e filtra i risultati prima di visualizzarli. Poiché l'intera HTTP risposta viene inviata al client prima del filtraggio, il filtraggio lato client può essere più lento del filtraggio lato server per set di dati di grandi dimensioni.

[L'interrogazione utilizza la sintassi per creare espressioni per filtrare l'output. JMESPath](#) Per imparare la JMESPath sintassi, consultate il [Tutorial](#) sul sito web. JMESPath

### Important

Il tipo di output specificato modifica il funzionamento dell'`--query` opzione:

- Se si specifica `--output text`, l'output viene impaginato prima dell'applicazione del `--query` filtro e quindi AWS CLI esegue la query una volta su ogni pagina dell'output. Per questo motivo, la query include il primo elemento corrispondente in ogni pagina, il che può

generare un output aggiuntivo inaspettato. Per filtrare ulteriormente l'output, puoi utilizzare altri strumenti da riga di comando come `head` o `tail`.

- Se si specifica `--output json` l'output viene completamente elaborato come un'unica struttura nativa prima dell'applicazione del `--query` filtro. AWS CLI Esegue la query solo una volta sull'intera struttura, producendo un risultato filtrato che viene quindi emesso.

## Argomenti di filtraggio lato client

- [Prima di iniziare](#)
- [Identificatori](#)
- [Selezione da un elenco](#)
- [Filtraggio dei dati annidati](#)
- [Risultati di appiattimento](#)
- [Filtraggio per valori specifici](#)
- [Espressioni di tubazione](#)
- [Filtraggio per più valori identificativi](#)
- [Aggiungere etichette ai valori degli identificatori](#)
- [Funzioni](#)
- [Esempi avanzati --query](#)

## Prima di iniziare

Quando utilizzate le espressioni di filtro utilizzate in questi esempi, assicuratevi di utilizzare le regole di quotazione corrette per la shell del terminale. Per ulteriori informazioni, consulta [the section called "Citazioni con stringhe"](#).

L'JSONOutput seguente mostra un esempio di ciò che il `--query` parametro può produrre. L'output descrive tre EBS volumi Amazon collegati a EC2 istanze Amazon separate.

## Output di esempio

```
$ 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
}
]
}

```

## Identificatori

Gli identificatori sono le etichette per i valori di output. Quando si creano filtri, si utilizzano gli identificatori per restringere i risultati delle query. Nel seguente esempio di output, tutti gli identificatori come `VolumesAvailabilityZone`, e `AttachTime` sono evidenziati.

```

$ 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
}
]
```

Per ulteriori informazioni, vedete [Identificatori](#) sul JMESPath sito Web.

## Selezione da un elenco

Un elenco o una matrice è un identificatore seguito da una parentesi quadra "[" come `Volumes` e `Attachments` in. [the section called "Prima di iniziare"](#)

### Sintassi

```
<listName>[ ]
```

Per filtrare tutto l'output di un array, è possibile utilizzare la notazione con caratteri jolly. Le espressioni [con caratteri jolly](#) sono espressioni utilizzate per restituire elementi utilizzando la notazione. \*

L'esempio seguente esegue una query su tutto il contenuto. `Volumes`

```
$ 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
}
]

```

Per visualizzare un volume specifico nell'array per indice, si chiama l'indice dell'array. Ad esempio, il primo elemento dell'`Volumes` array ha un indice pari a 0, che risulta nella `Volumes[0]` query. Per ulteriori informazioni sugli indici delle matrici, consultate le [espressioni di indice](#) sul JMESPath sito Web.

```

$ 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
}

```

Per visualizzare un intervallo specifico di volumi per indice, utilizzatelo `slice` con la seguente sintassi, dove `start` è l'indice iniziale dell'array, `stop` è l'indice in cui il filtro interrompe l'elaborazione e `step` è l'intervallo di salto.

## Sintassi

```
<arrayName>[<start>:<stop>:<step>]
```

Se uno di questi valori viene omissso dall'espressione `slice`, utilizza i seguenti valori predefiniti:

- Inizio: il primo indice dell'elenco, 0.
- Stop: l'ultimo indice dell'elenco.
- Fase: nessun passaggio da saltare, dove il valore è 1.

Per restituire solo i primi due volumi, utilizzate un valore iniziale pari a 0, un valore di arresto pari a 2 e un valore `Step` pari a 1, come illustrato nell'esempio seguente.

```
$ 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
}
]

```

Poiché questo esempio contiene valori predefiniti, potete abbreviare la sezione da `aVolumes[0:2:1].Volumes[:2]`

L'esempio seguente omette i valori predefiniti e restituisce ogni due volumi dell'intero array.

```

$ 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
  }
]

```

I passaggi possono anche utilizzare numeri negativi per filtrare nell'ordine inverso di una matrice, come illustrato nell'esempio seguente.

```

$ 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
  }
]

```

Per ulteriori informazioni, [consultate Slices](#) sul JMESPathsito Web.

## Filtraggio dei dati annidati

Per restringere il filtraggio dei `Volumes[*]` valori nidificati, utilizzate le sottoespressioni aggiungendo un punto e i criteri di filtro.

## Sintassi

```
<expression>.<expression>
```

L'esempio seguente mostra tutte le `Attachments` informazioni per tutti i volumi.

```

$ 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"
  }
]
]
```

Per filtrare ulteriormente i valori nidificati, aggiungete l'espressione per ogni identificatore nidificato. L'esempio seguente elenca il for all. State Volumes

```
$ aws ec2 describe-volumes \
  --query 'Volumes[*].Attachments[*].State'
[
  [
    "attached"
  ],
  [
    "attached"
  ]
]
```

```

],
[
  "attached"
]
]

```

## Risultati di appiattimento

Per ulteriori informazioni, consultate il [SubExpressions](#) JMESPath sito Web.

È possibile appiattare i risultati `Volumes[*].Attachments[*].State` rimuovendo la notazione jolly risultante dalla query. `Volumes[*].Attachments[].State` L'appiattimento spesso è utile per migliorare la leggibilità dei risultati.

```

$ aws ec2 describe-volumes \
  --query 'Volumes[*].Attachments[].State'
[
  "attached",
  "attached",
  "attached"
]

```

Per ulteriori informazioni, vedere [Flatten](#) sul sito Web. JMESPath

## Filtraggio per valori specifici

Per filtrare valori specifici in un elenco, si utilizza un'espressione di filtro come illustrato nella sintassi seguente.

### Sintassi

```
? <expression> <comparator> <expression>]
```

I comparatori di espressioni includono `==`, `!=`, `<`, `<=>`, e `>=`. L'esempio seguente filtra per `VolumeIds` for all `Volumes` in an `AttachedState`.

```

$ aws ec2 describe-volumes \
  --query 'Volumes[*].Attachments[?State==`attached`].VolumeId'
[
  [
    "vol-e11a5288"
  ],

```

```
[
  "vol-2e410a47"
],
[
  "vol-a1b3c7nd"
]
]
```

Questo può quindi essere appiattito, ottenendo l'esempio seguente.

```
$ aws ec2 describe-volumes \
  --query 'Volumes[*].Attachments[?State==`attached`].VolumeId[]'
[
  "vol-e11a5288",
  "vol-2e410a47",
  "vol-a1b3c7nd"
]
```

L'esempio seguente filtra tutti quelli Volumes che hanno una dimensione inferiore a 20. VolumeIds

```
$ aws ec2 describe-volumes \
  --query 'Volumes[?Size < `20`].VolumeId'
[
  "vol-2e410a47",
  "vol-a1b3c7nd"
]
```

Per ulteriori informazioni, consulta [Filter Expressions](#) sul JMESPath sito Web.

## Espressioni di tubazione

È possibile reindirizzare i risultati di un filtro a un nuovo elenco e quindi filtrare il risultato con un'altra espressione utilizzando la sintassi seguente:

## Sintassi

```
<expression> | <expression>]
```

L'esempio seguente prende i risultati del filtro dell'`Volumes[*].Attachments[].InstanceId` espressione e restituisce il primo risultato nell'array.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Attachments[].InstanceId | [0]'  
"i-a071c394"
```

Questo esempio esegue questa operazione creando innanzitutto l'array dalla seguente espressione.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Attachments[].InstanceId'  
"i-a071c394",  
"i-4b41a37c",  
"i-1jd73kv8"
```

Quindi restituisce il primo elemento di quell'array.

```
"i-a071c394"
```

Per ulteriori informazioni, consulta [Pipe Expressions](#) sul JMESPath sito Web.

Filtraggio per più valori identificativi

Per filtrare più identificatori, si utilizza un elenco a selezione multipla utilizzando la seguente sintassi:

Sintassi

```
<listName>[].[<expression>, <expression>]
```

Nell'esempio seguente, VolumeId e VolumeType vengono filtrati nell'Volumeselenco risultante dalla seguente espressione.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[][VolumeId, VolumeType]'  
[  
  [  
    "vol-e11a5288",  
    "standard"  
  ],  
  [  
    "vol-2e410a47",  
    "standard"  ]  
]
```

```

],
[
  "vol-a1b3c7nd",
  "standard"
]
]

```

Per aggiungere dati annidati all'elenco, aggiungete un altro elenco a selezione multipla. L'esempio seguente amplia l'esempio precedente filtrando anche per InstanceId e nell'elenco nidificato. State Attachments Il risultato è la seguente espressione.

```

$ 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"
      ]
    ]
  ]
]

```

```
]
]
```

Per renderla più leggibile, appiattite l'espressione come illustrato nell'esempio seguente.

```
$ 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"
  ]
]
```

Per ulteriori informazioni, vedere Elenco a selezione [multipla sul sito](#) Web. JMESPath

Aggiungere etichette ai valori degli identificatori

Per facilitare la lettura di questo output, utilizzate un hash a selezione multipla con la seguente sintassi.

Sintassi

```
<listName>[].{<label>: <expression>, <label>: <expression>}
```

L'etichetta identificativa non deve necessariamente corrispondere al nome dell'identificatore. L'esempio seguente utilizza l'etichetta `VolumeType` per i `VolumeType` valori.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[].{VolumeType: VolumeType}'  
[  
  {  
    "VolumeType": "standard",  
  },  
  {  
    "VolumeType": "standard",  
  },  
  {  
    "VolumeType": "standard",  
  }  
]
```

Per semplicità, l'esempio seguente mantiene i nomi degli identificatori per ogni etichetta e visualizza `VolumeId`, `VolumeType`, `InstanceId`, e `State` per tutti i volumi:

```
$ 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"  
  }  
]
```

Per ulteriori informazioni, consulta [Multiselect hash](#) sul sito Web. JMESPath

## Funzioni

La JMESPath sintassi contiene molte funzioni che è possibile utilizzare per le query. Per informazioni sulle JMESPath funzioni, consulta [Funzioni integrate sul JMESPath sito Web](#).

Per dimostrare come incorporare una funzione nelle query, l'esempio seguente utilizza la `sort_by` funzione. La `sort_by` funzione ordina un array utilizzando un'espressione come chiave di ordinamento utilizzando la sintassi seguente:

### Sintassi

```
sort_by(<listName>, <sort expression>)[<expression>
```

L'esempio seguente utilizza il precedente esempio di [hash multiselect](#) e ordina l'output per `VolumeId`

```
$ 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"
  }
]
```

[Per ulteriori informazioni, vedere `sort\_by` sul sito Web. JMESPath](#)



## Esempi avanzati --query

Per estrarre informazioni da un elemento specifico

L'esempio seguente utilizza il parametro `--query` per trovare un determinato elemento in un elenco e quindi estrae le informazioni da tale elemento. L'esempio elenca tutti gli endpoint di servizio `AvailabilityZones` associati all'endpoint di servizio specificato. Estrae l'elemento dall'elenco `ServiceDetails` che ha specificato `ServiceName`, quindi emette il campo `AvailabilityZones` da questo elemento selezionato.

```
$ 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"
  ]
]
```

Per mostrare istantanee dopo la data di creazione specificata

L'esempio seguente mostra come creare un elenco di tutte le snapshot che sono state create dopo una data specifica, includendo solo alcuni dei campi disponibili nell'output.

```
$ 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
  }
]
```

Per mostrare le più recenti AMIs

L'esempio seguente elenca le cinque Amazon Machine Images (AMIs) più recenti che hai creato, ordinate dalla più recente alla meno recente.

```
$ 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"
  }
]
```

Per mostrare istanze di Auto Scaling non integre

L'esempio seguente mostra solo le InstanceId istanze non integre nel gruppo Auto Scaling specificato.

```
$ aws autoscaling describe-auto-scaling-groups \
  --auto-scaling-group-name My-AutoScaling-Group-Name \
  --output text \
  --query 'AutoScalingGroups[*].Instances[?HealthStatus==`Unhealthy`].InstanceId'
```

Per includere i volumi con il tag specificato

L'esempio seguente descrive tutte le istanze con un `test` tag. Finché è presente un altro tag accanto `test` al volume, il volume viene comunque restituito nei risultati.

L'espressione seguente per restituire tutti i tag con il `test` tag in un array. Tutti i tag che non sono il `test` tag contengono un `null` valore.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Tags[?Value == `test`]'
```

Per escludere i volumi con il tag specificato

L'esempio seguente descrive tutte le istanze senza `test` tag. L'uso di `?Value != `test`` un'espressione semplice non funziona per escludere un volume poiché i volumi possono avere più tag. Finché c'è un altro tag `test` accanto al volume, il volume viene comunque restituito nei risultati.

Per escludere tutti i volumi con il `test` tag, inizia con l'espressione seguente per restituire tutti i tag con il `test` tag in un array. Tutti i tag che non sono il `test` tag contengono un `null` valore.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[*].Tags[?Value == `test`]'
```

Quindi filtra tutti i `test` risultati positivi utilizzando la `not_null` funzione.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[!not_null(Tags[?Value == `test`].Value)]'
```

Indirizza i risultati per appiattare i risultati risultanti dalla seguente query.

```
$ aws ec2 describe-volumes \  
  --query 'Volumes[!not_null(Tags[?Value == `test`].Value)] | []'
```

## Combinazione del filtraggio lato server e lato client

È possibile utilizzare il filtraggio lato server e lato client insieme. Per prima cosa viene completato il filtraggio lato server, che invia i dati al client che il parametro filtra quindi. `--query` Se utilizzi set di dati di grandi dimensioni, utilizzando innanzitutto il filtro lato server è possibile ridurre la quantità di dati inviati al client per ogni AWS CLI chiamata, pur mantenendo la potente personalizzazione fornita dal filtro lato client.

L'esempio seguente elenca EC2 i volumi Amazon che utilizzano filtri lato server e lato client. Il servizio filtra un elenco di tutti i volumi collegati nella zona di disponibilità. `us-west-2a` Il parametro `--query` limita ulteriormente l'output ai soli volumi con un valore `Size` maggiore di 50 e mostra solo i campi specificati con nomi definiti dall'utente.

```
$ 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"
  }
]
```

L'esempio seguente consente di recuperare un elenco di immagini che soddisfano diversi criteri. Quindi utilizza il parametro `--query` per ordinare l'output in base a `CreationDate`, selezionando solo il più recente. Infine, visualizza l'`ImageId` di tale immagine.

```
$ aws ec2 describe-images \
  --owners amazon \
  --filters "Name=name,Values=amzn*gp2" "Name=virtualization-type,Values=hvm" \
  "Name=root-device-type,Values=efs" \
  --query "sort_by(Images, &CreationDate)[-1].ImageId" \
  --output text
ami-00ced3122871a4921
```

L'esempio seguente visualizza il numero di volumi disponibili superiori a 1000 IOPS utilizzando `length` per contare quanti sono presenti in un elenco.

```
$ aws ec2 describe-volumes \
  --filters "Name=status,Values=available" \
  --query 'length(Volumes[?Iops > `1000`])'
3
```

## Risorse aggiuntive

### JMESPathTerminale

JMESPathTerminal è un comando terminale interattivo per sperimentare JMESPath espressioni utilizzate per il filtraggio lato client. Utilizzando il `jpterm` comando, il terminale mostra i risultati delle query immediati durante la digitazione. È possibile reindirizzare l' AWS CLI output direttamente al terminale, abilitando la sperimentazione avanzata di interrogazioni.

L'esempio seguente reindirizza l'`aws ec2 describe-volumes` output direttamente a Terminal. JMESPath

```
$ aws ec2 describe-volumes | jpterm
```

Per ulteriori informazioni su JMESPath Terminal e istruzioni di installazione, consulta [JMESPathTerminal](#) on GitHub.

### utilità jq

L'`jq` utilità offre un modo per trasformare l'output sul lato client nel formato di output desiderato. Per ulteriori informazioni `jq` e istruzioni di installazione, vedere [jq](#) on. GitHub

## Codici restituiti dalla riga di comando in AWS CLI

Il codice restituito è in genere un codice nascosto inviato dopo l'esecuzione di un comando AWS Command Line Interface (AWS CLI) che descrive lo stato del comando. È possibile utilizzare il `echo` comando per visualizzare il codice inviato dall'ultimo AWS CLI comando e utilizzare questi codici per determinare se un comando ha avuto esito positivo o negativo e perché un comando potrebbe avere un errore. Oltre ai codici restituiti, è possibile visualizzare ulteriori dettagli su un errore eseguendo i comandi con lo `--debug` switch. Questo switch produce un rapporto dettagliato dei passaggi AWS CLI utilizzati per elaborare il comando e del risultato di ogni passaggio.

Per determinare il codice restituito da un AWS CLI comando, esegui uno dei seguenti comandi subito dopo averlo eseguito. CLI

### Linux and macOS

```
$ echo $?  
0
```

## Windows PowerShell

```
PS> echo $lastexitcode  
0
```

## Windows Command Prompt

```
C:\> echo %errorlevel%  
0
```

Di seguito sono riportati i valori del codice restituito che possono essere restituiti al termine dell'esecuzione di un comando AWS Command Line Interface (AWS CLI).

Codic	Significato
0	Il servizio ha risposto con un codice di stato della HTTP risposta pari a 200, che indica che non sono stati generati errori dal AWS servizio AWS CLI e a cui è stata inviata la richiesta.
1	Una o più operazioni di trasferimento Amazon S3 non sono riuscite. Limitato ai comandi S3.
2	Il significato di questo codice restituito dipende dal comando. <ul style="list-style-type: none"><li>• Applicabile a tutti AWS CLI i comandi: il comando immesso non può essere analizzato. Gli errori di analisi possono essere causati, a titolo esemplificativo, dall'assenza di sottocomandi o argomenti richiesti oppure dall'utilizzo di comandi o argomenti sconosciuti.</li><li>• Limitato ai comandi S3: uno o più file contrassegnati per il trasferimento sono stati ignorati durante il processo di trasferimento. Tuttavia, tutti gli altri file contrassegnati per il trasferimento sono stati trasferiti. I file che vengono ignorati durante il processo di trasferimento includono: file che non esistono, file che sono dispositivi a caratteri speciali, dispositivi a blocchi, FIFO code o socket speciali e file per i quali l'utente non dispone delle autorizzazioni di lettura.</li></ul>
130	Il comando è stato interrotto da un. SIGINT Questo è il segnale inviato da te per annullare un comando con <code>Ctrl +C</code> .

Codic	Significato
255	Il comando ha avuto esito negativo. Si sono verificati errori generati dal AWS CLI o dal AWS servizio a cui è stata inviata la richiesta.

## Creazione e utilizzo di alias in AWS CLI

Gli alias sono scorciatoie che è possibile creare in AWS Command Line Interface (AWS CLI) per abbreviare i comandi o gli script utilizzati di frequente. Gli alias vengono creati nel `alias` file che si trova nella cartella di configurazione.

### Argomenti

- [Prerequisiti](#)
- [Fase 1: Creazione del file alias](#)
- [Fase 2: Creazione di un alias](#)
- [Fase 3: Chiamare un alias](#)
- [Esempi di repository di alias](#)
- [Risorse](#)

## Prerequisiti

Per utilizzare i comandi alias, è necessario completare quanto segue:

- Installa e configura la AWS CLI. Per ulteriori informazioni, consulta [Installare il AWS CLI](#) e [Credenziali di autenticazione e accesso per AWS CLI](#).
- Usa una AWS CLI versione minima di 1.11.24 o 2.0.0.
- (Facoltativo) Per utilizzare gli AWS CLI alias degli script bash, è necessario utilizzare un terminale compatibile con bash.

## Fase 1: Creazione del file alias

Per creare il `alias` file, è possibile utilizzare la navigazione dei file e un editor di testo oppure utilizzare il terminale preferito utilizzando la step-by-step procedura. Per creare rapidamente il tuo file alias, usa il seguente blocco di comandi.

## 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
```

### Per creare il file alias

1. Crea una cartella denominata `cli` nella tua cartella AWS CLI di configurazione. Per impostazione predefinita, la cartella di configurazione si trova `~/.aws/` su Linux o macOS e `%USERPROFILE%\.aws\` su Windows. È possibile crearla tramite la navigazione dei file o utilizzando il seguente comando.

## Linux and macOS

```
$ mkdir -p ~/.aws/cli
```

## Windows

```
C:\> md %USERPROFILE%\.aws\cli
```

Il percorso predefinito della `cli` cartella risultante è `~/.aws/cli/` su Linux o macOS e `%USERPROFILE%\.aws\cli` su Windows.

2. Nella `cli` cartella, create un file di testo denominato `alias` senza estensione e aggiungetelo `[toplevel]` alla prima riga. È possibile creare questo file tramite l'editor di testo preferito o utilizzare il seguente comando.

## Linux and macOS

```
$ echo '[toplevel]' > ~/.aws/cli/alias
```



## Windows

```
C:\> echo [toplevel] > %USERPROFILE%\.aws/cli/alias
```

## Fase 2: Creazione di un alias

È possibile creare un alias utilizzando i comandi di base o lo scripting bash.

### Creazione di un alias di comando di base

È possibile creare l'alias aggiungendo un comando utilizzando la seguente sintassi nel `alias` file creato nel passaggio precedente.

#### Sintassi

```
aliasname = command [--options]
```

Il **aliasname** è quello che chiamate il vostro alias. Il **command** è il comando da chiamare, che può includere altri alias. Puoi includere opzioni o parametri nel tuo alias o aggiungerli quando chiami il tuo alias.

L'esempio seguente crea un alias denominato `aws whoami` utilizzando il comando [aws sts get-caller-identity](#). Poiché questo alias chiama un AWS CLI comando esistente, è possibile scrivere il comando senza il `aws` prefisso.

```
whoami = sts get-caller-identity
```

L'esempio seguente riprende l'esempio precedente e aggiunge le opzioni di Account filtro e `testoutput`.

```
whoami2 = sts get-caller-identity --query Account --output text
```

## Creazione di un alias di sottocomando

### Note

La funzionalità degli alias dei sottocomandi richiede una AWS CLI versione minima di 1.11.24 o 2.0.0

È possibile creare un alias per i sottocomandi aggiungendo un comando utilizzando la seguente sintassi nel file creato nel passaggio precedente. `alias`

### Sintassi

```
[command commandGroup]  
aliasname = command [--options]
```

Il *commandGroup* è lo spazio dei nomi dei comandi, ad esempio il comando `aws ec2 describe-regions` trova nel gruppo di comandi. `ec2` Il *aliasname* è quello che chiami il tuo alias. Il *command* è il comando da chiamare, che può includere altri alias. Puoi includere opzioni o parametri nel tuo alias o aggiungerli quando chiami il tuo alias.

L'esempio seguente crea un alias denominato `aws ec2 regions` utilizzando il comando. [aws ec2 describe-regions](#) Poiché questo alias chiama un AWS CLI comando esistente nello spazio dei nomi del `ec2` comando, è possibile scrivere il comando senza il prefisso. `aws ec2`

```
[command ec2]  
regions = describe-regions --query Regions[].RegionName
```

Per creare alias da comandi esterni allo spazio dei nomi dei comandi, aggiungete un punto esclamativo al comando completo. L'esempio seguente crea un alias denominato utilizzando il comando. `aws ec2 instance-profiles` [aws iam list-instance-profiles](#)

```
[command ec2]  
instance-profiles = !aws iam list-instance-profiles
```

**Note**

Gli alias utilizzano solo gli spazi dei nomi di comando esistenti e non è possibile crearne di nuovi. Ad esempio, non è possibile creare un alias con la [command johnsmith] sezione poiché lo spazio dei nomi dei johnsmith comandi non esiste già.

## Creazione di un alias di scripting bash

**Warning**

Per utilizzare AWS CLI gli alias bash, è necessario utilizzare un terminale compatibile con bash

È possibile creare un alias utilizzando gli script bash per processi più avanzati utilizzando la seguente sintassi.

### Sintassi

```
aliasname =  
    !f() {  
        script content  
    }; f
```

Il *aliasname* è quello che chiami il tuo alias e *script content* è lo script che vuoi eseguire quando chiami l'alias.

L'esempio seguente lo utilizza `opendns` per visualizzare l'indirizzo IP corrente. Poiché è possibile utilizzare alias in altri alias, il seguente `myip` alias è utile per consentire o revocare l'accesso all'indirizzo IP dall'interno di altri alias.

```
myip =  
    !f() {  
        dig +short myip.opendns.com @resolver1.opendns.com  
    }; f
```

Il seguente esempio di script chiama l'`aws myipalias` precedente per autorizzare il tuo indirizzo IP per l'ingresso di un gruppo EC2 di sicurezza Amazon.

```
authorize-my-ip =
  !f() {
    ip=$(aws myip)
    aws ec2 authorize-security-group-ingress --group-id ${1} --cidr $ip/32 --protocol
tcp --port 22
  }; f
```

Quando chiami alias che utilizzano lo scripting bash, le variabili vengono sempre passate nell'ordine in cui le hai inserite. Nello scripting bash, i nomi delle variabili non vengono presi in considerazione, ma solo l'ordine in cui appaiono. Nel seguente esempio di `textalert` alias, la variabile per l'opzione è la prima e `--message` l'opzione è la seconda `--phone-number`.

```
textalert =
  !f() {
    aws sns publish --message "${1}" --phone-number ${2}
  }; f
```

## Fase 3: Chiamare un alias

Per eseguire l'alias creato nel `alias` file, utilizza la seguente sintassi. Puoi aggiungere opzioni aggiuntive quando chiami il tuo alias.

Sintassi

```
$ aws aliasname
```

L'esempio seguente utilizza il `aws whoami` comando alias.

```
$ aws
whoami
{
  "UserId": "A12BCD34E5FGHI6JKLM",
  "Account": "1234567890987",
  "Arn": "arn:aws:iam::1234567890987:user/userName"
}
```

L'esempio seguente utilizza l'`aws whoamialias` con opzioni aggiuntive per restituire solo il Account numero in `text` uscita.

```
$ aws whoami --query Account --output
text
1234567890987
```

L'esempio seguente utilizza l'alias del `aws ec2 regions` [sottocomando](#).

```
$ aws ec2
regions
[
  "ap-south-1",
  "eu-north-1",
  "eu-west-3",
  "eu-west-2",
  ...
```

## Chiamare un alias usando variabili di scripting bash

Quando si chiamano alias che utilizzano lo scripting bash, le variabili vengono passate nell'ordine in cui vengono inserite. Nello scripting bash, il nome delle variabili non viene preso in considerazione, ma solo l'ordine in cui appaiono. Ad esempio, nel seguente `textAlert` alias, la variabile per l'opzione `--message` è la prima ed `--phone-number` è la seconda.

```
textAlert =
!f() {
  aws sns publish --message "${1}" --phone-number ${2}
}; f
```

Quando si chiama l'`textAlert` alias, è necessario passare le variabili nello stesso ordine in cui vengono eseguite nell'alias. Nell'esempio seguente utilizziamo le variabili `$message` e `$phone`. La `$message` variabile viene passata come `${1}` per l'`--message` opzione e la `$phone` variabile viene passata come `${2}` per l'`--phone-number` opzione. Ciò comporta la corretta chiamata all'`textAlert` alias per inviare un messaggio.

```
$ aws textAlert $message
$phone
{
  "MessageId": "1ab2cd3e4-fg56-7h89-i01j-2klmn34567"
}
```

Nell'esempio seguente, l'ordine viene cambiato quando si chiama l'alias a and. \$phone \$message  
La \$phone variabile viene passata come \${1} per l'--messageopzione e la \$message variabile viene passata come \${2} per l'--phone-numberopzione. Poiché le variabili non sono ordinate, l'alias le passa in modo errato. Ciò causa un errore perché il contenuto di \$message non corrisponde ai requisiti di formattazione del numero di telefono per l'opzione. --phone-number

```
$ 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
```

## Esempi di repository di alias

Il [repository di AWS CLI alias GitHub](#) contiene esempi di AWS CLI alias creati dal team di sviluppatori e dalla AWS CLI community. È possibile utilizzare l'intero esempio di alias file o utilizzare singoli alias per uso personale.

### Warning

L'esecuzione dei comandi in questa sezione elimina il file esistente alias. Per evitare di sovrascrivere il file alias esistente, modifica la posizione di download.

Per utilizzare gli alias del repository

1. Installa Git. Per istruzioni di installazione, vedi [Guida introduttiva - Installazione di Git](#) nella documentazione di Git.
2. Installa il jpm comando. Il jpm comando viene utilizzato nell'tostringalias. Per le istruzioni di installazione, consultate il file [JMESPath\(jp\) README .md on. GitHub](#)
3. Installa il comando. jq Il jq comando viene utilizzato nell'tostring-with-jqalias. Per le istruzioni di installazione, vedete il [JSONprocessore \(jq\) acceso](#). GitHub
4. Scaricate il alias file effettuando una delle seguenti operazioni:

- Esegui i seguenti comandi che vengono scaricati dal repository e copia il `alias` file nella cartella di configurazione.

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
```

- Scaricalo direttamente dal repository e salvalo nella cartella della tua `cli` cartella di AWS CLI configurazione. Per impostazione predefinita, la cartella di configurazione si trova `~/.aws/` su Linux o macOS e `%USERPROFILE%\aws\` su Windows.
5. Per verificare che gli alias funzionino, esegui il seguente alias.

```
$ aws whoami
```

Viene visualizzata la stessa risposta del `aws sts get-caller-identity` comando:

```
{
  "Account": "012345678901",
  "UserId": "AIUAINBADX2VEG2TC6HD6",
  "Arn": "arn:aws:iam::012345678901:user/myuser"
}
```

## Risorse

- Il [repository di AWS CLI alias GitHub](#) contiene esempi di AWS CLI alias creati dal team di AWS CLI sviluppatori e dal contributo della community. AWS CLI
- L'annuncio della funzionalità alias tratto da [AWS re:Invent 2016: The Effective User on. AWS CLI](#) YouTube
- [aws sts get-caller-identity](#)

- [aws ec2 describe-instances](#)
- [aws sns publish](#)



# Esempi di codice per AWS CLI

Questo capitolo fornisce una raccolta di esempi che mostrano come usare il AWS Command Line Interface (AWS CLI) con Servizi AWS.

In AWS CLI questa guida sono riportati i seguenti tipi di esempi:

- [Esempi di comandi guidati](#) - Esempi di comandi guidati per la Guida per AWS CLI l'utente su come utilizzarli AWS CLI con some Servizi AWS. Questi sono spesso esempi più dettagliati rispetto agli esempi della [guida AWS CLI di riferimento AWS CLI](#) .
- [AWS CLI esempi di comandi](#) - Esempi di comandi open source disponibili anche nella [guida di AWS CLI riferimento AWS CLI](#) . Gli esempi di comandi sono ospitati nel [AWS CLI repository](#) su GitHub.
- [AWS CLI utilizzando esempi di codice di scripting Bash - Esempi di script bash open source](#). [Gli esempi di script di Bash sono ospitati nel Code Examples Repository su AWS GitHub](#)

## Esempio di feedback

Non riesci a trovare quello che ti serve? Richiedi un esempio di comando utilizzando il link Fornisci feedback nella parte inferiore di questa pagina o nella pagina di comando pertinente nella [guida di AWS CLI riferimento AWS CLI](#) .

Vuoi contribuire? Contribuisci con esempi di AWS CLI comandi nel [AWS Code Examples Repository](#) su GitHub. Per ulteriori informazioni su come contribuire, consultate i [passaggi rapidi sulle GitHub pagine relative ai contributi di esempio di AWS CLI codice](#).

## Esempi di comandi guidati per AWS CLI

Il AWS Command Line Interface (AWS CLI) è uno strumento open source che consente di interagire con un'ampia gamma di comandi di Servizi AWS utilizzo nella shell della riga di comando. Questa sezione fornisce esempi guidati che mostrano come sfruttare il file per accedere AWS CLI ad alcuni di. Servizi AWS Ciò include alcuni AWS CLI comandi personalizzati, come i comandi di alto livello `aws s3`. Questi esempi di comandi illustrano le azioni comuni utilizzate per alcuni Servizi AWS e forniscono risorse aggiuntive per ulteriori informazioni.

Che siate AWS utenti esperti o alle prime armi AWS CLI, questi esempi guidati rappresentano una risorsa per semplificare le AWS operazioni.

Per un riferimento completo di tutti i comandi disponibili per ciascuno di essi Servizio AWS, consulta la [guida AWS CLI di riferimento AWS CLI versione 2 della guida](#) . Inoltre, puoi utilizzare la [guida integrata nella riga di comando](#) per esplorare la serie di comandi Servizi AWS, opzioni e funzionalità di AWS CLI.

Per altri esempi di comandi non disponibili in questa sezione, consultate la [AWS CLI esempi di comandi](#) sezione. Questi sono esempi di comandi open source disponibili anche nella [guida di AWS CLI riferimento AWS CLI](#) . Gli esempi di comandi sono ospitati nel [AWS CLI](#) repository su GitHub.

Per esempi di script bash open source, vedere. [the section called “esempi di script Bash”](#) Gli esempi di script in Bash sono ospitati nel [AWS Code](#) Examples Repository su. GitHub

## Servizi

- [Utilizzo di Amazon DynamoDB in AWS CLI](#)
- [Utilizzo di Amazon EC2 in AWS CLI](#)
- [Utilizzo di Amazon S3 Glacier in AWS CLI](#)
- [Utilizzo IAM in AWS CLI](#)
- [Utilizzo di Amazon S3 in AWS CLI](#)
- [Accesso ad Amazon SNS in AWS CLI](#)

## Utilizzo di Amazon DynamoDB in AWS CLI

Un'introduzione ad Amazon DynamoDB

[Cos'è Amazon DynamoDB?](#)

Il AWS Command Line Interface (AWS CLI) fornisce supporto per tutti i servizi di AWS database, incluso Amazon DynamoDB. Puoi utilizzarlo AWS CLI per operazioni improvvisate, come la creazione di una tabella. È inoltre possibile utilizzarlo per incorporare le operazioni DynamoDB negli script di utilità.

Per ulteriori informazioni sull'utilizzo di AWS CLI con DynamoDB, [dynamodb](#) vedere nel AWS CLI Command Reference.

Per elencare i AWS CLI comandi per DynamoDB, utilizzare il seguente comando.

```
$ aws dynamodb help
```

## Argomenti

- [Prerequisiti](#)
- [Creazione e utilizzo di tabelle DynamoDB](#)
- [Utilizzo di DynamoDB Local](#)
- [Risorse](#)

## Prerequisiti

Per eseguire i dynamodb comandi, è necessario:

- Installa e configura la AWS CLI. Per ulteriori informazioni, consulta [Installare il AWS CLI](#) e [Credenziali di autenticazione e accesso per AWS CLI](#).

## Creazione e utilizzo di tabelle DynamoDB

Il formato della riga di comando è costituito da un nome di comando DynamoDB, seguito dai parametri per quel comando. AWS CLI Supporta la sintassi CLI [abbreviata per i valori dei parametri e la sintassi](#) completa. JSON

L'esempio seguente crea una tabella denominata MusicCollection.

```
$ 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
```

Puoi aggiungere nuove righe alla tabella con comandi simili a quelli mostrati nell'esempio seguente. Questi esempi utilizzano una combinazione di sintassi abbreviata e. 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"
  }
}

```

Può essere difficile comporre un comando valido JSON in una sola riga. Per semplificare questa operazione, AWS CLI possono leggere JSON i file. Ad esempio, si consideri il seguente JSON frammento, che è memorizzato in un file denominato `expression-attributes.json`

```

{
  ":v1": {"S": "No One You Know"},
  ":v2": {"S": "Call Me Today"}
}

```

Puoi utilizzare questo file per inviare una richiesta query con l' AWS CLI. Nell'esempio seguente, il contenuto del file `expression-attributes.json` viene utilizzato come valore del parametro `--expression-attribute-values`.

```
$ 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  
}
```

## Utilizzo di DynamoDB Local

Oltre a DynamoDB, puoi utilizzarlo AWS CLI con DynamoDB Local. DynamoDB Local è un piccolo database e server lato client che imita il servizio DynamoDB. DynamoDB Local consente di scrivere applicazioni che utilizzano DynamoDB, senza manipolare tabelle o dati nel servizio web API DynamoDB. Al contrario, tutte le azioni vengono reindirizzate a un database locale API. In questo modo è possibile salvare il throughput assegnato, lo storage dei dati e le tariffe per il trasferimento dei dati.

Per ulteriori informazioni su DynamoDB Local e su come utilizzarlo con, consulta le seguenti sezioni AWS CLI della Amazon [DynamoDB](#) Developer Guide:

- [DynamoDB Local](#)
- [Utilizzo di AWS CLI con DynamoDB Local](#)

## Risorse

AWS CLI riferimento:

- [aws dynamodb](#)
- [aws dynamodb create-table](#)
- [aws dynamodb put-item](#)
- [aws dynamodb query](#)

Riferimento del servizio:

- [DynamoDB Local nella Amazon DynamoDB Developer Guide](#)
- [Utilizzo di AWS CLI con DynamoDB Local nella Amazon DynamoDB Developer Guide](#)

## Utilizzo di Amazon EC2 in AWS CLI

Un'introduzione ad Amazon Elastic Compute Cloud

[Introduzione ad Amazon EC2 - Elastic Cloud Server e hosting con AWS](#)

Amazon Elastic Compute Cloud (AmazonEC2) offre ambienti di elaborazione virtuale altamente scalabili e flessibili. Amazon EC2 consente di fornire e gestire server virtuali, noti come EC2 istanze Amazon, per soddisfare un'ampia gamma di esigenze di elaborazione.

Le EC2 istanze Amazon sono macchine virtuali che possono essere personalizzate con varie configurazioni di memoriaCPU, archiviazione e funzionalità di rete. Puoi scegliere tra una vasta gamma di tipi di istanze, che vanno da opzioni leggere ed economiche a istanze potenti e ad alte prestazioni, a seconda dei requisiti dell'applicazione. Questa flessibilità consente di soddisfare le esigenze di elaborazione per ottimizzare le prestazioni e l'efficienza dei costi.

Inoltre, Amazon EC2 offre una suite di funzionalità che ti consentono di gestire le tue risorse di elaborazione in modo efficace. Queste includono la possibilità di avviare rapidamente nuove istanze, creare immagini di macchine personalizzate (AMIs) per una rapida implementazione e aumentare o ridurre la capacità di elaborazione in base alle esigenze.

Puoi accedere alle funzionalità di Amazon EC2 utilizzando AWS Command Line Interface (AWS CLI). Per elencare i AWS CLI comandi per AmazonEC2, usa il seguente comando.

```
aws ec2 help
```

Prima di eseguire qualsiasi comando, imposta le tue credenziali di default. Per ulteriori informazioni, consulta [Configurazione delle impostazioni per AWS CLI](#).

Questo argomento mostra brevi esempi di AWS CLI comandi che eseguono attività comuni per AmazonEC2.

Per esempi di AWS CLI comandi in formato esteso, consulta il [repository degli esempi di AWS CLI codice](#) su. GitHub

### Argomenti

- [Creazione, visualizzazione ed eliminazione di coppie di EC2 chiavi Amazon in AWS CLI](#)
- [Creazione, configurazione ed eliminazione di gruppi di EC2 sicurezza Amazon in AWS CLI](#)
- [Avvio, pubblicazione e chiusura di EC2 istanze Amazon nel AWS CLI](#)
- [Modifica un tipo di EC2 istanza Amazon con uno script bash nel AWS CLI](#)

## Creazione, visualizzazione ed eliminazione di coppie di EC2 chiavi Amazon in AWS CLI

Puoi usare AWS Command Line Interface (AWS CLI) per creare, visualizzare ed eliminare le tue coppie di chiavi per Amazon Elastic Compute Cloud (AmazonEC2). Utilizzi coppie di chiavi per connetterti a un'EC2istanza Amazon.

È necessario fornire la coppia di chiavi ad Amazon EC2 quando si crea l'istanza e quindi utilizzare tale coppia di chiavi per l'autenticazione quando ci si connette all'istanza.

### Note

Per ulteriori esempi di comandi, consulta la [AWS CLI guida AWS CLI](#) .

### Argomenti

- [Prerequisiti](#)
- [Creazione di una coppia di chiavi](#)
- [Visualizzazione della coppia di chiavi](#)
- [Eliminazione della coppia di chiavi](#)
- [Riferimenti](#)

## Prerequisiti

Per eseguire i `ec2` comandi, è necessario:

- Installa e configura la AWS CLI. Per ulteriori informazioni, consulta [Installare il AWS CLI](#) e [Credenziali di autenticazione e accesso per AWS CLI](#).
- Imposta le tue IAM autorizzazioni per consentire l'EC2 accesso ad Amazon. Per ulteriori informazioni sulle IAM autorizzazioni per Amazon EC2, consulta [IAM le politiche per Amazon EC2 nella Amazon EC2 User Guide](#).

## Creazione di una coppia di chiavi

Per creare una coppia di chiavi, utilizzate il `aws ec2 create-key-pair` comando con l'`--query` opzione e l'`--output text` opzione per reindirizzare la chiave privata direttamente in un file.

```
$ aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text
> MyKeyPair.pem
```

Per impostazione predefinita PowerShell, il `>` file reindirizzamento è la codifica UTF -8, che non può essere utilizzata con alcuni client. SSH Pertanto, devi convertire l'output reindirizzandolo al comando `out-file` e impostando esplicitamente la codifica su `ascii`.

```
PS C:\>aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text
| out-file -encoding ascii -filepath MyKeyPair.pem
```

L'aspetto del file `MyKeyPair.pem` risultante è simile a quello riportato di seguito.

```
-----BEGIN RSA PRIVATE KEY-----
EXAMPLEKEYKCAQEAY7WZhaDsrA1W3mRlQtvhwY0RRX8gnxgDAfRt/gx42kWXsT4rXE/b5CpSgie/
vBoU7jLxx92pNHoFnByP+Dc21eyyz6CvjTmWA0JwfWiW5/akH7i05dSrvC7dQkW2duV5QuUdEQW
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
YkriL0DbLX1vRAH+yHPRit2hH0jtUNZ4Aaxv+cpq09qbUI3+43eEy24B7G/Uh+GTfbjsXs0xQx/x
p9otyVwc7hsQ5TA5PZb+mvkJ50BEKzet9XcKw0NBVELGhnEPe7cCgYEA06Vgov6YHleHui9kHuws
```



```

ayav0e1c5zkxjF9nfHFJRry21R1trw2Vdpn+9g481URipzWV0Eihvm+xTtmaZ1Sp//1kq75XDwnU
WA8gkn603QE3fq2yN98BURsAKdJfJ5RL1HvGQvTe10HLYYXpJnEkHv+Unl2ajLivWUt5pbBrKbUC
gYBjb0+0Zk0sCcpZ29sbzjYjpIddErySIyRX5gV2uNQwAjLdp9PfN295yQ+BxMBXiIycWVQiw0bH
oMo7yykABY70zd5wQewBQ4AdS1WSX4nGDtsiFxiI5sKuAAe0CbTosy1s8w8fxoJ5Tz1sdoxNeGs
Arq6Wv/G16zQuAE9zK9vwwKBgF+09VI/1wJBirsDGz9whVwfFPtTkJNvJZzYt69qezx1sjgFKshy
WBhd4xHZtmCqpBP1AymEjr/T01bxyARmXMnIOWIANNXMGB4KGSy11mzSVAoQ+fqR+cJ3d0dyP11j
jjb0Ed/NY8fr1NDxAVHE8BSkdsx2f6ELEyBKJSRr9snRAoGAMrTwYneXzvTskF/S5Fyu0i0egLDa
NWUH38v/nDCgEpIXD5Hn3qAEcju1IjmbwlvTW+nY2jVhv7UGd8MjwUTNGItbdb6nsYqM2asrnF3qS
VRkAKKKYeGjKpUfVTtW0YFjXkfcR/V+QFL50ndHAKJXjW7a4ejJLncTzmZSpYzwApc=
-----END RSA PRIVATE KEY-----

```

La tua chiave privata non è memorizzata AWS e può essere recuperata solo al momento della creazione. Non puoi recuperarla successivamente. Se perdi la chiave privata, invece, devi creare una nuova coppia di chiavi.

Se ti connetti all'istanza da un computer Linux, consigliamo di utilizzare il comando seguente per impostare le autorizzazioni del file della chiave privata in modo da essere l'unico a poterlo leggere.

```
$ chmod 400 MyKeyPair.pem
```

### Visualizzazione della coppia di chiavi

Una «impronta digitale» viene generata dalla tua coppia di chiavi e puoi usarla per verificare che la chiave privata che hai sul tuo computer locale corrisponda alla chiave pubblica archiviata in AWS.

L'impronta digitale è un SHA1 hash ricavato da una copia DER codificata della chiave privata. Questo valore viene acquisito al momento della creazione della coppia di chiavi e memorizzato AWS con la chiave pubblica. Puoi visualizzare l'impronta digitale nella EC2 console Amazon o eseguendo il AWS CLI comando [aws ec2 describe-key-pairs](#).

L'esempio seguente mostra l'impronta per 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"
    }
  ]
}

```

Per ulteriori informazioni su chiavi e impronte digitali, consulta [Amazon EC2 Key Pairs](#) nella Amazon EC2 User Guide.

## Eliminazione della coppia di chiavi

Per eliminare una key pair, esegui il [aws ec2 delete-key-pair](#) comando, sostituendo *MyKeyPair* con il nome della coppia da eliminare.

```
$ aws ec2 delete-key-pair --key-name MyKeyPair
```

## Riferimenti

AWS CLI riferimento:

- [aws ec2](#)
- [aws ec2 create-key-pair](#)
- [aws ec2 delete-key-pair](#)
- [aws ec2 describe-key-pairs](#)

Altro riferimento:

- [Amazon Elastic Compute Cloud Documentation](#)
- Per visualizzare e contribuire a AWS SDK esempi di AWS CLI codice, consulta il [AWS Code Examples Repository](#) su GitHub.

## Creazione, configurazione ed eliminazione di gruppi di EC2 sicurezza Amazon in AWS CLI

Puoi creare un gruppo di sicurezza per le tue istanze Amazon Elastic Compute Cloud (AmazonEC2) che funge essenzialmente da firewall, con regole che determinano il traffico di rete che può entrare e uscire.

Usa AWS Command Line Interface (AWS CLI) per creare un gruppo di sicurezza, aggiungere regole ai gruppi di sicurezza esistenti ed eliminare gruppi di sicurezza.

### Note

Per ulteriori esempi di comandi, consultate la [AWS CLI guida AWS CLI](#).

## Argomenti

- [Prerequisiti](#)
- [Creazione di un gruppo di sicurezza](#)
- [Aggiunta di regole al gruppo di sicurezza](#)
- [Eliminazione del gruppo di sicurezza:](#)
- [Riferimenti](#)

## Prerequisiti

Per eseguire i `ec2` comandi, è necessario:

- Installa e configura la AWS CLI. Per ulteriori informazioni, consulta [Installare il AWS CLI e Credenziali di autenticazione e accesso per AWS CLI](#).
- Imposta le tue IAM autorizzazioni per consentire l'EC2 accesso ad Amazon. Per ulteriori informazioni sulle IAM autorizzazioni per AmazonEC2, consulta [IAM le politiche per Amazon EC2 nella Amazon EC2 User Guide](#).

## Creazione di un gruppo di sicurezza

Puoi creare gruppi di sicurezza associati a cloud privati virtuali (VPCs).

L'[aws ec2 create-security-group](#) esempio seguente mostra come creare un gruppo di sicurezza per una determinata area VPC.

```
$ aws ec2 create-security-group --group-name my-sg --description "My security group" --  
vpc-id vpc-1a2b3c4d  
{  
  "GroupId": "sg-903004f8"  
}
```

Per visualizzare le informazioni iniziali per un gruppo di sicurezza, esegui il [aws ec2 describe-security-groups](#) comando. È possibile fare riferimento a un gruppo VPC di sicurezza EC2 - solo in base al nome `vpc-id`, non in base al nome.

```
$ 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"
}
]
```

## Aggiunta di regole al gruppo di sicurezza

Quando esegui un'istanza Amazon EC2, devi abilitare le regole nel gruppo di sicurezza per consentire il traffico di rete in entrata per i tuoi mezzi di connessione all'immagine.

Ad esempio, se stai avviando un'istanza Windows, in genere aggiungi una regola per consentire al traffico in entrata sulla TCP porta 3389 di supportare Remote Desktop Protocol (RDP). Se stai lanciando un'istanza Linux, in genere aggiungi una regola per consentire al traffico in entrata sulla TCP porta 22 di supportare le connessioni SSH.

Utilizza il comando [aws ec2 authorize-security-group-ingress](#) per aggiungere una regola al gruppo di sicurezza. Un parametro richiesto di questo comando è l'indirizzo IP pubblico del computer o la rete (sotto forma di intervallo di indirizzi) a cui è collegato il computer, in [CIDR](#) notazione.

### Note

Forniamo il seguente servizio, <https://checkip.amazonaws.com/>, per consentirti di determinare il tuo indirizzo IP pubblico. Per trovare altri servizi che possono aiutarti a identificare il tuo indirizzo IP, utilizza il browser per cercare "qual è il indirizzo IP". Se ti connetti tramite ISP o da un firewall protetto utilizzando un indirizzo IP dinamico (tramite un NAT gateway di una

rete privata), l'indirizzo può cambiare periodicamente. In questo caso, devi scoprire l'intervallo di indirizzi IP utilizzato dai computer client.

L'esempio seguente mostra come aggiungere una regola per RDP (TCPporta 3389) a un EC2 gruppo di VPC sicurezza con l'ID `sg-903004f8` utilizzando il proprio indirizzo IP.

Per iniziare, trova il tuo indirizzo IP.

```
$ curl https://checkip.amazonaws.com  
x.x.x.x
```

Puoi quindi aggiungere l'indirizzo IP al tuo gruppo di sicurezza eseguendo il [aws ec2 authorize-security-group-ingress](#) comando.

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port 3389 --cidr x.x.x.x/x
```

Il comando seguente aggiunge un'altra regola da abilitare SSH alle istanze dello stesso gruppo di sicurezza.

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port 22 --cidr x.x.x.x/x
```

Per visualizzare le modifiche al gruppo di sicurezza, esegui il [aws ec2 describe-security-groups](#) comando.

```
$ 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"
}
]
}

```

Eliminazione del gruppo di sicurezza:

Per eliminare un gruppo di sicurezza, esegui il [aws ec2 delete-security-group](#) comando.

#### Note

Non puoi eliminare un gruppo di sicurezza se è attualmente collegato a un ambiente.

Il seguente esempio di comando elimina un gruppo VPC di sicurezza EC2 -.

```
$ aws ec2 delete-security-group --group-id sg-903004f8
```

#### Riferimenti

AWS CLI riferimento:

- [aws ec2](#)
- [aws ec2 authorize-security-group-ingress](#)

- [aws ec2 create-security-group](#)
- [aws ec2 delete-security-group](#)
- [aws ec2 describe-security-groups](#)

Altro riferimento:

- [Amazon Elastic Compute Cloud Documentation](#)
- Per visualizzare e contribuire a AWS SDK esempi di AWS CLI codice, consulta il [AWS Code Examples Repository](#) su GitHub.

## Avvio, pubblicazione e chiusura di EC2 istanze Amazon nel AWS CLI

Puoi utilizzare AWS Command Line Interface (AWS CLI) per avviare, elencare e terminare le istanze di Amazon Elastic Compute Cloud EC2 (Amazon). Se avvii un'istanza che non rientra nel piano AWS gratuito, ti verrà fatturata una volta avviata l'istanza e ti verrà addebitato il tempo in cui l'istanza è in esecuzione, anche se rimane inattiva.

### Note

Per ulteriori esempi di comandi, consulta la [guida AWS CLI di AWS CLI](#)

## Argomenti

- [Prerequisiti](#)
- [Lancio dell'istanza](#)
- [Aggiunta di un dispositivo a blocchi all'istanza](#)
- [Aggiunta di un tag all'istanza](#)
- [Connessione all'istanza](#)
- [Elencazione delle istanze](#)
- [Interruzione di un'istanza](#)
- [Riferimenti](#)

## Prerequisiti

Per eseguire i ec2 comandi descritti in questo argomento, è necessario:

- Installa e configura la AWS CLI. Per ulteriori informazioni, consulta [Installare il AWS CLI e Credenziali di autenticazione e accesso per AWS CLI](#).
- Imposta le tue IAM autorizzazioni per consentire l'EC2accesso ad Amazon. Per ulteriori informazioni sulle IAM autorizzazioni per AmazonEC2, consulta [IAMle politiche per Amazon EC2 nella Amazon EC2 User Guide](#).
- Crea una [key pair](#) e un [gruppo di sicurezza](#).
- Seleziona un'Amazon Machine Image (AMI) e annota l'AMIID. Per ulteriori informazioni, consulta [Finding a Suitable AMI](#) in the Amazon EC2 User Guide.

## Lancio dell'istanza

Per avviare un'EC2istanza Amazon utilizzando il comando AMI selezionato, utilizza il [aws ec2 run-instances](#) comando. Puoi avviare l'istanza in un cloud privato virtuale (VPC).

Inizialmente l'istanza viene visualizzata in stato pending, ma dopo pochi minuti passa allo stato running.

L'esempio seguente mostra come avviare un't2.microistanza nella sottorete specificata di aVPC. Sostituire il *italicized* valori dei parametri con i propri.

```
$ 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
}
]
```

```
}
```

## Aggiunta di un dispositivo a blocchi all'istanza

A ogni istanza avviata è associato un volume dispositivo root. Puoi utilizzare la mappatura dei dispositivi a blocchi per specificare volumi Amazon Elastic Block Store (AmazonEBS) aggiuntivi o volumi di instance store da collegare a un'istanza al momento del lancio.

Per aggiungere un dispositivo a blocchi all'istanza, specifica l'opzione `--block-device-mappings` quando utilizzi `run-instances`.

Il seguente parametro di esempio fornisce un EBS volume Amazon standard di 20 GB e lo mappa alla tua istanza utilizzando l'identificatore `/dev/sdf`.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\",\"Ebs\":{\"VolumeSize\":20, \"DeleteOnTermination\":false} }]"
```

L'esempio seguente aggiunge un EBS volume Amazon, mappato a `/dev/sdf`, basato su uno snapshot esistente. Una snapshot rappresenta un'immagine caricata sul volume per te. Quando specifichi una snapshot, non devi specificare una dimensione del volume perché sarà sufficiente a contenere la tua immagine. Tuttavia, se specifichi la dimensione, questa deve essere maggiore o uguale a quella della snapshot.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\",\"Ebs\":{\"SnapshotId\":\"snap-a1b2c3d4\"} }]"
```

L'esempio seguente aggiunge due volumi all'istanza. Il numero di volumi disponibili per l'istanza dipende dal tipo di istanza.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\",\"VirtualName\":\"ephemeral0\"}, {\"DeviceName\":\"/dev/sdg\",\"VirtualName\":\"ephemeral1\"}]"
```

L'esempio seguente crea la mappatura (`/dev/sdj`), ma non esegue il provisioning di un volume per l'istanza.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdj\",\"NoDevice\":\"\"}]"
```

Per ulteriori informazioni, consulta [Block Device Mapping](#) nella Amazon EC2 User Guide.

## Aggiunta di un tag all'istanza

Un tag è un'etichetta che assegni a una AWS risorsa. Questo consente di aggiungere metadati alle risorse, che possono essere utilizzati per vari scopi. Per ulteriori informazioni, consulta [Tagging Your Resources](#) nella Amazon EC2 User Guide.

L'esempio seguente mostra come aggiungere un tag con il nome chiave "Name" e il valore "MyInstance" all'istanza specificata, utilizzando il [aws ec2 create-tags](#) comando.

```
$ aws ec2 create-tags --resources i-5203422c --tags Key=Name,Value=MyInstance
```

## Connessione all'istanza

Mentre l'istanza è in esecuzione, puoi connetterti e utilizzarla esattamente come se si trovasse nel tuo computer. Per ulteriori informazioni, consulta [Connect to Your Amazon EC2 Instance](#) nella Amazon EC2 User Guide.

## Elencazione delle istanze

Puoi usare il AWS CLI per elencare le tue istanze e visualizzare le informazioni su di esse. Puoi creare un elenco di tutte le istanze oppure filtrare i risultati in base alle istanze che ti interessano.

Negli esempi seguenti viene illustrato come utilizzare il [aws ec2 describe-instances](#) comando.

Il comando seguente elenca tutte le istanze.

```
$ aws ec2 describe-instances
```

Il comando seguente filtra l'elenco visualizzando solo le istanze t2.micro e restituisce solo i valori InstanceId per ogni corrispondenza.

```
$ aws ec2 describe-instances --filters "Name=instance-type,Values=t2.micro" --query  
"Reservations[].Instances[].InstanceId"  
[  
  "i-05e998023d9c69f9a"  
]
```

Il comando seguente elenca qualsiasi istanza con il tag Name=MyInstance.

```
$ aws ec2 describe-instances --filters "Name=tag:Name,Values=MyInstance"
```

Il comando seguente elenca le istanze che sono state avviate utilizzando uno dei seguenti comandi AMIs: `ami-x0123456`, `ami-y0123456`, e `ami-z0123456`

```
$ aws ec2 describe-instances --filters "Name=image-id,Values=ami-x0123456,ami-y0123456,ami-z0123456"
```

## Interruzione di un'istanza

Se termini un'istanza, questa viene eliminata. Dopo averla terminata, non puoi più connetterti a un'istanza.

Appena lo stato dell'istanza cambia in `shutting-down` o `terminated`, vengono bloccati i rispettivi addebiti. Per riconnetterti a un'istanza in un secondo momento, utilizza [stop-instances](#) invece di `terminate-instances`. Per ulteriori informazioni, consulta [Terminate Your Instance](#) nella Amazon EC2 User Guide.

Per eliminare un'istanza, usi il comando [aws ec2 terminate-instances](#) per eliminarla.

```
$ aws ec2 terminate-instances --instance-ids i-5203422c
{
  "TerminatingInstances": [
    {
      "InstanceId": "i-5203422c",
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

## Riferimenti

AWS CLI riferimento:

- [aws ec2](#)
- [aws ec2 create-tags](#)

- [aws ec2 describe-instances](#)
- [aws ec2 run-instances](#)
- [aws ec2 terminate-instances](#)

Altro riferimento:

- [Amazon Elastic Compute Cloud Documentation](#)
- Per visualizzare e contribuire a AWS SDK esempi di AWS CLI codice, consulta il [AWS Code Examples Repository](#) su GitHub.

## Modifica un tipo di EC2 istanza Amazon con uno script bash nel AWS CLI

Questo esempio di script bash per Amazon EC2 modifica il tipo di istanza per un'EC2istanza Amazon utilizzando AWS Command Line Interface (AWS CLI). Arresta l'istanza se è in esecuzione, cambia il tipo di istanza e quindi, se richiesto, riavvia l'istanza. Gli script di shell sono programmi progettati per essere eseguiti in un'interfaccia a riga di comando.

### Note

Per ulteriori esempi di comandi, consultate la [AWS CLI guida AWS CLI](#).

## Argomenti

- [Prima di iniziare](#)
- [Informazioni su questo esempio](#)
- [Parametri](#)
- [File](#)
- [Riferimenti](#)

## Prima di iniziare

Prima di poter eseguire uno qualsiasi degli esempi seguenti, è necessario completare le seguenti operazioni.

- Installa e configura la AWS CLI. Per ulteriori informazioni, consulta [Installare il AWS CLI](#) e [Credenziali di autenticazione e accesso per AWS CLI](#).

- Il profilo utilizzato deve disporre di autorizzazioni che consentano le AWS operazioni eseguite dagli esempi.
- Un'EC2istanza Amazon in esecuzione nell'account per la quale sei autorizzato a interrompere e modificare. Se esegui lo script di test, quest'ultimo avvia automaticamente un'istanza, verifica la modifica del tipo e quindi termina l'istanza.
- Come procedura AWS ottimale, concedi a questo codice il minimo privilegio o solo le autorizzazioni necessarie per eseguire un'operazione. Per ulteriori informazioni, vedere [Grant Least Privilege](#) nella Guida per l'utente di AWS Identity and Access Management (IAM).
- Questo codice non è stato testato in tutte le AWS regioni. Alcuni AWS servizi sono disponibili solo in regioni specifiche. Per ulteriori informazioni, consulta [Service Endpoints and Quotas](#) nella Guida di riferimento AWS generale.
- L'esecuzione di questo codice può comportare addebiti sul tuo AWS account. È tua responsabilità assicurarti che tutte le risorse create da questo script vengano rimosse quando hai finito di utilizzarle.

### Informazioni su questo esempio

Questo esempio è scritto come una funzione nel file di script di shell `change_ec2_instance_type.sh` che è possibile utilizzare `source` da un altro script o dalla riga di comando. Ogni file di script contiene commenti che descrivono ciascuna delle funzioni. Una volta che la funzione è in memoria, è possibile richiamarla dalla riga di comando. Ad esempio, i seguenti comandi modificano il tipo dell'istanza specificata `int2`. `nano`:

```
$ source ./change_ec2_instance_type.sh
$ ./change_ec2_instance_type -i *instance-id* -t new-type
```

Per l'esempio completo e i file di script scaricabili, consulta [Change Amazon EC2 Instance Type](#) nel AWS Code Examples Repository su GitHub

### Parametri

`-i` - (stringa) Specifica l'ID dell'istanza da modificare.

`-t` - (stringa) Specifica il tipo di EC2 istanza Amazon a cui passare.

`-r` - (switch) Per impostazione predefinita, non è impostato. Se `-r` è impostato, riavvia l'istanza dopo il cambio di tipo.

**-f** - (switch) Per impostazione predefinita, lo script richiede all'utente di confermare la chiusura dell'istanza prima di effettuare il passaggio. Se **-f** è impostata, la funzione non richiede all'utente di cambiare tipo prima di chiudere l'istanza

**-v** - (switch) Per impostazione predefinita, lo script funziona silenziosamente e visualizza l'output solo in caso di errore. Se **-v** è impostata, la funzione visualizza lo stato durante tutto il suo funzionamento.

File

## **change\_ec2\_instance\_type.sh**

Il file di script principale contiene la `change_ec2_instance_type()` funzione che esegue le seguenti attività:

- Verifica l'esistenza dell'EC2istanza Amazon specificata.
- A meno che non **-f** sia selezionata, avvisa l'utente prima di interrompere l'istanza.
- Cambia il tipo di istanza
- Se impostata **-r**, riavvia l'istanza e conferma che l'istanza è in esecuzione

Visualizza il codice per [change\\_ec2\\_instance\\_type.sh](#) on. GitHub

## **test\_change\_ec2\_instance\_type.sh**

`test_change_ec2_instance_type.sh` Lo script del file verifica i vari percorsi di codice per la `change_ec2_instance_type` funzione. Se tutti i passaggi dello script di test funzionano correttamente, lo script di test rimuove tutte le risorse che ha creato.

È possibile eseguire lo script di test con i seguenti parametri:

- **-v** - (switch) Ogni test mostra uno pass/failure status as they run. By default, the tests runs silently and the output includes only the final overall pass/failure stato.
- **-i** - (switch) Lo script viene messo in pausa dopo ogni test per consentire all'utente di sfogliare i risultati intermedi di ogni passaggio. Consente di esaminare lo stato corrente dell'istanza utilizzando la EC2 console Amazon. Lo script procede al passaggio successivo dopo aver premuto ENTER al prompt.

Visualizza il codice per [test\\_change\\_ec2\\_instance\\_type.sh](#) on. GitHub



## awsdocs\_general.sh

Il file di script `awsdocs_general.sh` contiene funzioni generiche utilizzate in esempi avanzati per AWS CLI.

Visualizza il codice per [awsdocs\\_general.sh](#) on GitHub.

### Riferimenti

AWS CLI riferimento:

- [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](#)

Altro riferimento:

- [Amazon Elastic Compute Cloud Documentation](#)
- Per visualizzare e contribuire a AWS SDK esempi di AWS CLI codice, consulta il [AWS Code Examples Repository](#) su GitHub.

## Utilizzo di Amazon S3 Glacier in AWS CLI

Un'introduzione ad Amazon S3 Glacier

[Introduzione ad Amazon S3 Glacier](#)

Questo argomento mostra esempi di AWS CLI comandi che eseguono attività comuni per S3 Glacier. Gli esempi mostrano come utilizzare AWS CLI per caricare un file di grandi dimensioni su S3 Glacier suddividendolo in parti più piccole e caricandole dalla riga di comando.

Puoi accedere alle funzionalità di Amazon S3 Glacier utilizzando (). AWS Command Line Interface  
AWS CLI Per elencare i AWS CLI comandi per S3 Glacier, usa il seguente comando.

```
aws glacier help
```

### Note

Per riferimenti ai comandi ed esempi aggiuntivi, [aws glacier](#) consultate la sezione Command Reference.AWS CLI

## Argomenti

- [Prerequisiti](#)
- [Crea un vault Amazon S3 Glacier](#)
- [Preparazione di un file per il caricamento](#)
- [Avvio di un caricamento in più parti e caricamento dei file](#)
- [Completamento del caricamento](#)
- [Risorse](#)

## Prerequisiti

Per eseguire i glacier comandi, è necessario:

- Installa e configura la AWS CLI. Per ulteriori informazioni, consulta [Installare il AWS CLI](#) e [Credenziali di autenticazione e accesso per AWS CLI](#).
- Questo tutorial utilizza diversi strumenti a riga di comando che in genere vengono preinstallati su sistemi operativi di tipo Unix, tra cui Linux e macOS. Gli utenti Windows possono utilizzare gli stessi strumenti installando [Cygwin](#) ed eseguendo i comandi dal terminale Cygwin. Le utilità e i comandi nativi di Windows che eseguono le stesse funzioni sono indicati, laddove disponibili.

## Crea un vault Amazon S3 Glacier

Creazione di un insieme di credenziali con il comando [create-vault](#).

```
$ aws glacier create-vault --account-id - --vault-name myvault
```

```
{  
  "location": "/123456789012/vaults/myvault"  
}
```

### Note

Tutti i comandi S3 Glacier richiedono un parametro ID account. Utilizza il trattino (`--account-id -`) per utilizzare l'account corrente.

## Preparazione di un file per il caricamento

Creazione di un file per il caricamento di prova. I seguenti comandi creano un file denominato *largefile* che contiene esattamente 3 MiB di dati casuali.

### Linux o 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` è un'utilità che consente di copiare una serie di byte da un file di input a un file di output. L'esempio precedente usa il file del dispositivo di sistema `/dev/urandom` come origine dei dati casuali. `fsutil` esegue una funzione simile in Windows.

### Windows

```
C:\> fsutil file createnew largefile 3145728  
File C:\temp\largefile is created
```

Successivamente, dividi il file in blocchi da 1 MiB (1.048.576 byte) utilizzando uno splitter di file.

```
$ split -b 1048576 --verbose largefile chunk  
creating file `chunkaa'  
creating file `chunkab'  
creating file `chunkac'
```

## Avvio di un caricamento in più parti e caricamento dei file

Crea un caricamento multiparte in Amazon S3 Glacier utilizzando il comando. [initiate-multipart-upload](#)

```
$ aws glacier initiate-multipart-upload --account-id - --archive-description "multipart upload test" --part-size 1048576 --vault-name myvault
{
  "uploadId": "19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-0ssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ",
  "location": "/123456789012/vaults/myvault/multipart-uploads/19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-0ssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ"
}
```

S3 Glacier richiede la dimensione di ogni parte in byte (1 MiB in questo esempio), il nome del vault e un ID account per configurare il caricamento multiparte. AWS CLI Emette un ID di caricamento quando l'operazione è completa. Salva l'ID in una variabile shell da utilizzare in futuro.

Linux o macOS

```
$ UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-0ssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ"
```

Windows

```
C:\> set UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-0ssZtLqyFu7sY1_1R7vgFuJV6NtcV5zpsJ"
```

Quindi, utilizza il comando [upload-multipart-part](#) per caricare singolarmente le tre parti.

```
$ 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

L'esempio precedente utilizza il simbolo del dollaro (\$) per fare riferimento ai contenuti della variabile della shell UPLOADID in Linux. Nella riga di comando di Windows, utilizzate il segno di percentuale (%) su entrambi i lati del nome della variabile (ad esempio,%UPLOADID%).

È necessario specificare l'intervallo di byte di ogni parte al momento del caricamento in modo che S3 Glacier possa riassemblarla nell'ordine corretto. Ogni parte è 1048576 byte, pertanto la prima parte occupa i byte 0-1048575, la seconda i byte 1048576-2097151 e la terza i byte 2097152-3145727.

## Completamento del caricamento

Amazon S3 Glacier richiede un tree hash del file originale per confermare che tutte le parti caricate siano rimaste intatte. AWS

Per calcolare un hash ad albero, è necessario dividere il file in parti da 1 MiB e calcolare un hash SHA binario -256 per ogni pezzo. Quindi si suddivide l'elenco di hash in coppie, si combinano i due hash binari in ciascuna coppia e si utilizzano gli hash risultanti. Ripeti il processo fino a esaurire tutti gli hash. Se il numero di hash è dispari a qualsiasi livello, promuovilo al livello successivo senza modificarlo.

La chiave per calcolare correttamente una struttura hash quando si usano le utility a riga di comando consiste nel memorizzare ogni hash in formato binario ed effettuare la conversione in esadecimale solo nell'ultima fase. Combinando ed eseguendo l'hashing della versione esadecimale di qualsiasi hash nella struttura si otterrà un risultato non corretto.

### Note

Gli utenti Windows possono utilizzare il comando type anziché cat. [Open SSL è disponibile per Windows all'indirizzo Open.org. SSL](#)

## Per calcolare una struttura hash

1. Se non lo hai già fatto, suddividi il file originale in parti da 1 MiB.

```
$ split --bytes=1048576 --verbose largefile chunk
creating file `chunkaa'
creating file `chunkab'
creating file `chunkac'
```

2. Calcola e memorizza l'hash binario SHA -256 di ogni blocco.

```
$ openssl dgst -sha256 -binary chunkaa > hash1
$ openssl dgst -sha256 -binary chunkab > hash2
$ openssl dgst -sha256 -binary chunkac > hash3
```

3. Combina i primi due hash e utilizza l'hash binario del risultato.

```
$ cat hash1 hash2 > hash12
$ openssl dgst -sha256 -binary hash12 > hash12hash
```

4. Combina l'hash padre dei blocchi aa e ab con l'hash del blocco ac ed esegui l'hashing del risultato, questa volta con un output esadecimale. Memorizza il risultato in una variabile della shell.

```
$ cat hash12hash hash3 > hash123
$ openssl dgst -sha256 hash123
SHA256(hash123)= 9628195fcdcbbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
$ TREEHASH=9628195fcdcbbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
```

Infine, completa il caricamento con il comando [complete-multipart-upload](#). Questo comando utilizza le dimensioni del file originale in byte, il valore hash nella struttura finale esadecimale, il tuo ID account e il nome dell'insieme di credenziali.

```
$ 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-
kWUyS1dSB1mgPPWkEbeFfqDSav053rU7FvVLHfRc6hg",
  "checksum": "9628195fcdcbbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
```

```
"location": "/123456789012/vaults/myvault/archives/
d3AbWhE0YE1m6f_fI1jPG82F8xzbMEEZmrALLGAA0NJAzo5QdP-N83MKqd96Unspoa5H51ItWX-sK8-
QS0ZhwsyGiu9-R-kwWUyS1dSB1mgPPWkEbeFfqDSav053rU7FvVLHfRc6hg"
}
```

Puoi anche controllare lo stato del vault utilizzando il comando [describe-vault](#).

```
$ 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

Lo stato dell'insieme di credenziali viene aggiornato circa una volta al giorno. Per ulteriori informazioni, consulta l'argomento relativo all'[utilizzo di vault](#).

Ora puoi rimuovere i file di blocco e hash creati in modo sicuro.

```
$ rm chunk* hash*
```

Per ulteriori informazioni sui caricamenti in più parti, consulta [Uploading Large Archives in Parts and Computing Checksums nella Amazon S3 Glacier Developer Guide](#).

## Risorse

AWS CLI riferimento:

- [aws glacier](#)
- [aws glacier complete-multipart-upload](#)
- [aws glacier create-vault](#)
- [aws glacier describe-vault](#)

- [aws glacier initiate-multipart-upload](#)

Riferimento del servizio:

- [Guida per sviluppatori di Amazon S3 Glacier](#)
- [Caricamento di archivi di grandi dimensioni in parti](#) nella Amazon S3 Glacier Developer Guide
- [Elaborazione dei checksum nella Guida per gli sviluppatori di Amazon S3 Glacier](#)
- [Utilizzo dei Vaults nella Guida](#) per sviluppatori di Amazon S3 Glacier

## Utilizzo IAM in AWS CLI

Un'introduzione a AWS Identity and Access Management

[Introduzione a AWS Identity and Access Management](#)

È possibile accedere alle funzionalità di AWS Identity and Access Management (IAM) utilizzando AWS Command Line Interface (AWS CLI). Per elencare i AWS CLI comandi per IAM, utilizzate il seguente comando.

```
aws iam help
```

Questo argomento mostra esempi di AWS CLI comandi che eseguono attività comuni per IAM.

Prima di eseguire qualsiasi comando, imposta le tue credenziali di default. Per ulteriori informazioni, consulta [Configurazione delle impostazioni per AWS CLI](#).

Per ulteriori informazioni sul IAM servizio, consulta la [Guida AWS Identity and Access Management per l'utente](#).

Argomenti

- [Creazione di utenti e gruppi IAM](#)
- [Allegare una politica IAM gestita a un utente](#)
- [Impostazione di una password iniziale per un utente IAM](#)
- [Creazione di una chiave di accesso per un IAM utente](#)



## Creazione di utenti e gruppi IAM

Per creare un gruppo e aggiungervi un nuovo utente

1. Utilizza il comando [create-group](#) per creare il gruppo.

```
$ 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. Utilizza il comando [create-user](#) per creare l'utente.

```
$ 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. Utilizza il comando [add-user-to-group](#) per aggiungere l'utente al gruppo.

```
$ aws iam add-user-to-group --user-name MyUser --group-name MyIamGroup
```

4. Per verificare che il gruppo MyIamGroup contenga MyUser, utilizza il comando [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"
}

```

## Allegare una politica IAM gestita a un utente

La policy nell'esempio fornisce all'utente "Power User Access" (Accesso utenti avanzati).

Per allegare una politica IAM gestita a un utente

1. Determina l'Amazon Resource Name (ARN) della policy da allegare. Il seguente comando utilizza `list-policies` per trovare ARN la politica con il nome `PowerUserAccess`. Quindi lo ARN memorizza in una variabile di ambiente.

```

$ export POLICYARN=$(aws iam list-policies --query 'Policies[?
PolicyName==`PowerUserAccess`].{ARN:Arn}' --output text) ~
$ echo $POLICYARN
arn:aws:iam::aws:policy/PowerUserAccess

```

2. Per allegare la politica, usa il [attach-user-policy](#) comando e fai riferimento alla variabile di ambiente che contiene la politica ARN.

```

$ aws iam attach-user-policy --user-name MyUser --policy-arn $POLICYARN

```

3. Verifica che la policy sia collegata all'utente eseguendo il comando [list-attached-user-policies](#).

```

$ aws iam list-attached-user-policies --user-name MyUser
{
  "AttachedPolicies": [
    {

```

```
        "PolicyName": "PowerUserAccess",
        "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"
    }
]
}
```

Per ulteriori informazioni, consulta la sezione relativa all'[accesso alle risorse di gestione](#). Questo argomento fornisce collegamenti a una panoramica delle autorizzazioni e delle politiche e collegamenti a esempi di politiche per l'accesso ad Amazon S3, EC2 Amazon e altri servizi.

## Impostazione di una password iniziale per un utente IAM

Il comando seguente consente [create-login-profile](#) di impostare una password iniziale per l'utente specificato. Quando l'utente accede per la prima volta, deve modificare la password in qualcosa che solo l'utente conosce.

```
$ 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
  }
}
```

È possibile utilizzare il `update-login-profile` comando per modificare la password di un utente.

```
$ aws iam update-login-profile --user-name MyUser --password My!User1ADifferentP@ssword
```

## Creazione di una chiave di accesso per un IAM utente

È possibile utilizzare il [create-access-key](#) comando per creare una chiave di accesso per un utente. Una chiave di accesso è un set di credenziali di sicurezza costituito da un ID chiave di accesso e una chiave segreta.

Un utente può creare solo due chiavi di accesso alla volta. Se provi a creare un terzo set, il comando restituisce un errore `LimitExceeded`.

```
$ 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"
  }
}
```

Usa il [delete-access-key](#) comando per eliminare una chiave di accesso per un utente. Specifica la chiave di accesso da eliminare utilizzando l'ID chiave di accesso.

```
$ aws iam delete-access-key --user-name MyUser --access-key-id AKIAIOSFODNN7EXAMPLE
```

## Utilizzo di Amazon S3 in AWS CLI

Un'introduzione ad Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3)

[Introduzione ad Amazon Simple Storage Service \(Amazon S3\) - Archiviazione nel cloud su AWS](#)

Puoi accedere alle funzionalità di Amazon Simple Storage Service (Amazon S3) utilizzando AWS Command Line Interface (). AWS CLI Amazon S3 è un servizio di storage di oggetti altamente scalabile e durevole. Amazon S3 è progettato per fornire una capacità di storage praticamente illimitata, il che lo rende una soluzione ideale per un'ampia gamma di esigenze di archiviazione e gestione dei dati.

Amazon S3 consente di archiviare e recuperare qualsiasi quantità di dati, da file di piccole dimensioni a set di dati di grandi dimensioni, sotto forma di oggetti. Ogni oggetto è archiviato in un contenitore chiamato bucket, a cui è possibile accedere e gestire tramite AWS Management Console o in modo programmatico tramite gli AWS SDKs strumenti e AWS CLI

Oltre allo storage di base, Amazon S3 offre anche una gamma di funzionalità tra cui gestione del ciclo di vita, controllo delle versioni, scalabilità e sicurezza. Queste si integrano con altre e ti Servizi AWS consentono di creare soluzioni basate sul cloud che si adattano alle tue esigenze.

AWS CLI Fornisce due livelli di comandi per accedere ad Amazon S3:

- **s3** — Comandi personalizzati di alto livello creati appositamente per semplificare l' AWS CLI esecuzione di attività comuni, come la creazione, la manipolazione, l'eliminazione e la sincronizzazione di oggetti e bucket.
- **s3api**: offre l'accesso diretto a tutte le operazioni di Amazon S3 che consente di eseguire API operazioni avanzate.

Argomenti di questa guida:

- [Utilizzo di comandi di alto livello \(s3\) in AWS CLI](#)
- [Utilizzo dei API comandi -Level \(s3api\) in AWS CLI](#)
- [Esempio di scripting per il ciclo di vita del bucket Amazon S3 nel AWS CLI](#)

## Utilizzo di comandi di alto livello (s3) in AWS CLI

Questo argomento descrive alcuni dei comandi che è possibile utilizzare per gestire i bucket e gli oggetti Amazon S3 utilizzando i [aws s3](#) comandi presenti in. AWS CLI Per i comandi non trattati in questo argomento e per altri esempi di comandi, consulta [aws s3](#) i comandi nella AWS CLI Guida di riferimento.

I `aws s3` comandi di alto livello semplificano la gestione degli oggetti Amazon S3. Questi comandi consentono di gestire i contenuti di Amazon S3 all'interno di se stesso e con le directory locali.

Argomenti

- [Prerequisiti](#)
- [Prima di iniziare](#)
- [Creazione di un bucket](#)
- [Elenca i bucket e gli oggetti](#)
- [Eliminazione di bucket](#)
- [Eliminazione di oggetti](#)
- [Spostare oggetti](#)
- [Copia oggetti](#)
- [Sincronizza oggetti](#)
- [Opzioni utilizzate di frequente per i comandi s3](#)
- [Risorse](#)

## Prerequisiti

Per eseguire i s3 comandi, devi:

- Installa e configura la AWS CLI. Per ulteriori informazioni, consulta [Installare il AWS CLI](#) e [Credenziali di autenticazione e accesso per AWS CLI](#).
- Il profilo utilizzato deve disporre di autorizzazioni che consentano le AWS operazioni eseguite dagli esempi.
- Comprendi questi termini di Amazon S3:
  - Bucket: cartella Amazon S3 di primo livello.
  - Prefisso: una cartella Amazon S3 in un bucket.
  - Oggetto: qualsiasi elemento ospitato in un bucket Amazon S3.

## Prima di iniziare

Questa sezione descrive alcune cose da tenere a mente prima di utilizzare `aws s3` i comandi.

### Caricamenti di oggetti di grandi dimensioni

Quando usi `aws s3` i comandi per caricare oggetti di grandi dimensioni in un bucket Amazon S3, esegue AWS CLI automaticamente un caricamento in più parti. Non puoi riprendere un caricamento non riuscito quando usi questi comandi. `aws s3`

Se il caricamento in più parti non riesce a causa di un timeout o se l'hai annullato manualmente in AWS CLI, AWS CLI interrompe il caricamento e ripulisce tutti i file che sono stati creati. Questo processo può richiedere alcuni minuti.

Se il processo di caricamento o pulizia in più parti viene annullato da un comando kill o da un errore di sistema, i file creati rimangono nel bucket Amazon S3. <https://docs.aws.amazon.com/cli/latest/reference/s3api/abort-multipart-upload.html>

### Proprietà e tag dei file nelle copie in più parti

Quando utilizzi la AWS CLI versione 1 dei comandi nello spazio dei `aws s3` nomi per copiare un file da una posizione del bucket Amazon S3 a un'altra posizione del bucket Amazon S3 e tale operazione [utilizza la copia multiparte](#), nessuna proprietà del file dall'oggetto di origine viene copiata nell'oggetto di destinazione.

## Creazione di un bucket

Utilizzate il [s3 mb](#) comando per creare un bucket. I nomi dei bucket devono essere univoci a livello globale (unici in tutto Amazon S3) e devono essere DNS conformi.

Possono includere lettere minuscole, numeri, trattini e punti. Possono iniziare e terminare solo con una lettera o un numero e non possono contenere un punto accanto a un trattino o a un altro periodo.

### Sintassi

```
$ aws s3 mb <target> [--options]
```

### esempi di s3 mb

L'esempio seguente crea il `s3://amzn-s3-demo-bucket` bucket.

```
$ aws s3 mb s3://amzn-s3-demo-bucket
```

## Elenca i bucket e gli oggetti

Per elencare i tuoi bucket, cartelle o oggetti, usa il [s3 ls](#) comando. Se si utilizza il comando senza destinazione o opzioni, vengono elencati tutti i bucket.

### Sintassi

```
$ aws s3 ls <target> [--options]
```

Per alcune opzioni comuni da utilizzare con questo comando ed esempi, vedere [Opzioni utilizzate di frequente per i comandi s3](#). Per un elenco completo delle opzioni disponibili, vedere [s3 ls](#) nella Guida ai AWS CLI comandi.

### esempi di s3 ls

L'esempio seguente elenca tutti i bucket Amazon S3.

```
$ 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
```

Il comando seguente elenca tutti gli oggetti e i prefissi in un bucket. In questo output di esempio, il prefisso `example/` ha un solo file denominato `MyFile1.txt`

```
$ aws s3 ls s3://amzn-s3-demo-bucket
                PRE example/
2018-12-04 19:05:48          3 MyFile1.txt
```

Puoi filtrare l'output per un determinato prefisso includendolo nel comando. Il comando seguente elenca gli oggetti in *bucket-name/example/* (ovvero, oggetti in *bucket-name* filtrato in base al prefisso *example/*).

```
$ aws s3 ls s3://amzn-s3-demo-bucket/example/
2018-12-06 18:59:32          3 MyFile1.txt
```

## Eliminazione di bucket

Per eliminare un bucket, usa il comando. [s3 rb](#)

### Sintassi

```
$ aws s3 rb <target> [--options]
```

### esempi s3 rb

L'esempio seguente rimuove il bucket. *s3://amzn-s3-demo-bucket*

```
$ aws s3 rb s3://amzn-s3-demo-bucket
```

Per default, perché l'operazione riesca il bucket deve essere vuoto. Per rimuovere un bucket che non è vuoto, è necessario includere l'opzione `--force`. Se stai usando un bucket con versione che contiene oggetti precedentemente eliminati, ma mantenuti, questo comando non ti consente di rimuovere il bucket. Devi prima rimuovere tutti i contenuti.

L'esempio seguente elimina tutti gli oggetti e i prefissi nel bucket, quindi elimina il bucket.

```
$ aws s3 rb s3://amzn-s3-demo-bucket --force
```

## Eliminazione di oggetti

Per eliminare oggetti in un bucket o nella directory locale, utilizzate il comando. [s3 rm](#)

### Sintassi



```
$ aws s3 rm <target> [--options]
```

Per alcune opzioni comuni da utilizzare con questo comando ed esempi, consulta [Opzioni utilizzate di frequente per i comandi s3](#). Per un elenco completo delle opzioni, vedere [s3 rm](#) nella Guida ai AWS CLI comandi.

esempi di rm s3

L'esempio seguente elimina `filename.txt` da `s3://amzn-s3-demo-bucket/example`

```
$ aws s3 rm s3://amzn-s3-demo-bucket/example/filename.txt
```

L'esempio seguente elimina tutti gli oggetti dall'`s3://amzn-s3-demo-bucket/example` utilizzo dell'`--recursive` opzione.

```
$ aws s3 rm s3://amzn-s3-demo-bucket/example --recursive
```

Spostare oggetti

Utilizzate il [s3 mv](#) comando per spostare oggetti da un bucket o da una directory locale. Il `s3 mv` comando copia l'oggetto o il file di origine nella destinazione specificata, quindi elimina l'oggetto o il file di origine.

Sintassi

```
$ aws s3 mv <source> <target> [--options]
```

Per alcune opzioni comuni da utilizzare con questo comando ed esempi, vedere [Opzioni utilizzate di frequente per i comandi s3](#). Per un elenco completo delle opzioni disponibili, vedere [s3 mv](#) nella Guida ai AWS CLI comandi.

#### Warning

Se utilizzi qualsiasi tipo di punto di accesso ARNs o alias del punto di accesso nella tua origine o URIs destinazione Amazon S3, devi prestare particolare attenzione che Amazon URIs S3 di origine e destinazione si risolvano in diversi bucket sottostanti. Se i bucket di origine e di destinazione coincidono, il file o l'oggetto di origine può essere spostato su se stesso, il che può comportare l'eliminazione accidentale del file o dell'oggetto di origine. Per verificare che i bucket di origine e di destinazione non siano

gli stessi, utilizzate il `--validate-same-s3-paths` parametro o impostate la variabile `AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS` di ambiente su `true`

esempi di `s3 mv`

L'esempio seguente sposta tutti gli oggetti da `s3://amzn-s3-demo-bucket/example`. `s3://amzn-s3-demo-bucket/`

```
$ aws s3 mv s3://amzn-s3-demo-bucket/example s3://amzn-s3-demo-bucket/
```

L'esempio seguente sposta un file locale dalla directory di lavoro corrente al bucket Amazon S3 con il comando. `s3 mv`

```
$ aws s3 mv filename.txt s3://amzn-s3-demo-bucket
```

L'esempio seguente sposta un file dal bucket Amazon S3 alla directory di lavoro corrente, dove `./` specifica la directory di lavoro corrente.

```
$ aws s3 mv s3://amzn-s3-demo-bucket/filename.txt ./
```

Copia oggetti

Usa il `s3 cp` comando per copiare oggetti da un bucket o da una directory locale.

Sintassi

```
$ aws s3 cp <source> <target> [--options]
```

È possibile utilizzare il parametro dash per lo streaming di file su standard input (`stdin`) o standard output (`stdout`).

#### Warning

Se lo stai usando PowerShell, la shell potrebbe alterare la codifica di a CRLF o aggiungere un input o output piped o CRLF reindirizzare l'output.

Il `s3 cp` comando utilizza la seguente sintassi per caricare un flusso di file da `stdin` un bucket specificato.

## Sintassi

```
$ aws s3 cp - <target> [--options]
```

Il `s3 cp` comando utilizza la seguente sintassi per scaricare un flusso di file Amazon S3 per `stdout`

## Sintassi

```
$ aws s3 cp <target> [--options] -
```

Per alcune opzioni comuni da usare con questo comando ed esempi, consulta [Opzioni utilizzate di frequente per i comandi s3](#). Per l'elenco completo delle opzioni, vedere [s3 cp](#) nella Guida ai AWS CLI comandi.

## Esempi di `s3 cp`

L'esempio seguente copia tutti gli oggetti da `s3://amzn-s3-demo-bucket/example` a `s3://amzn-s3-demo-bucket/`.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/example s3://amzn-s3-demo-bucket/
```

L'esempio seguente copia un file locale dalla directory di lavoro corrente al bucket Amazon S3 con il comando `s3 cp`

```
$ aws s3 cp filename.txt s3://amzn-s3-demo-bucket
```

L'esempio seguente copia un file dal bucket Amazon S3 nella directory di lavoro corrente, dove `./` specifica la directory di lavoro corrente.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/filename.txt ./
```

L'esempio seguente utilizza `echo` per trasmettere il testo «hello world» al file `s3://bucket-name/filename.txt`

```
$ echo "hello world" | aws s3 cp - s3://amzn-s3-demo-bucket/filename.txt
```

L'esempio seguente trasmette il `s3://amzn-s3-demo-bucket/filename.txt` file alla console `stdout` e ne stampa il contenuto.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/filename.txt -  
hello world
```

L'esempio seguente trasmette il contenuto di `s3://bucket-name/pre` to stdout, utilizza il `bzip2` comando per comprimere i file e carica il nuovo file compresso denominato `to.key.bz2` `s3://bucket-name`

```
$ aws s3 cp s3://amzn-s3-demo-bucket/pre - | bzip2 --best | aws s3 cp - s3://amzn-s3-  
demo-bucket/key.bz2
```

## Sincronizza oggetti

Il [s3 sync](#) comando sincronizza il contenuto di un bucket e una directory o il contenuto di due bucket. In genere, `s3 sync` copia file o oggetti mancanti o obsoleti tra l'origine e la destinazione. Tuttavia, puoi anche utilizzare l'opzione `--delete` per rimuovere dalla destinazione i file o gli oggetti non presenti nell'origine.

## Sintassi

```
$ aws s3 sync <source> <target> [--options]
```

Per alcune opzioni comuni da utilizzare con questo comando ed esempi, vedere [Opzioni utilizzate di frequente per i comandi s3](#). Per un elenco completo delle opzioni, vedere [s3 sync](#) nella Guida ai AWS CLI comandi.

## esempi di sincronizzazione s3

L'esempio seguente sincronizza il contenuto di un prefisso Amazon S3 denominato `path` nel bucket denominato `amzn-s3-demo-bucket` con la directory di lavoro corrente.

`s3 sync` aggiorna tutti i file che hanno una dimensione o un'ora di modifica diversi dai file con lo stesso nome nella destinazione. L'output mostra le operazioni specifiche eseguite durante la sincronizzazione. Si noti che l'operazione sincronizza in modo ricorsivo la sottodirectory `MySubdirectory` e il suo contenuto con. `s3://amzn-s3-demo-bucket/path/MySubdirectory`

```
$ aws s3 sync . s3://mamzn-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
```

L'esempio seguente, che estende quello precedente, mostra come utilizzare l'opzione. `--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
```

Quando si utilizza l'`--delete` opzione, le `--include` opzioni `--exclude` e possono filtrare file o oggetti da eliminare durante un'`s3 sync` operazione. In questo caso, la stringa di parametro deve specificare i file da escludere o includere per l'eliminazione nel contesto della directory o del bucket di destinazione. Di seguito viene riportato un esempio.

```
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
```

## Opzioni utilizzate di frequente per i comandi s3

Le seguenti opzioni vengono utilizzate di frequente per i comandi descritti in questo argomento. Per un elenco completo delle opzioni che è possibile utilizzare su un comando, vedere il comando specifico nella [guida di AWS CLI riferimento](#) alla .

### acl

s3 sync e s3 cp può usare l'--acl opzione. Ciò consente di impostare le autorizzazioni di accesso per i file copiati su Amazon S3. L'opzione --acl accetta i valori private, public-read e public-read-write. Per ulteriori informazioni, consulta [Canned ACL](#) nella Amazon S3 User Guide.

```
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --acl public-read
```

### escludere

Quando si utilizza il s3 rm comando s3 cp, s3 mv, s3 sync, o, è possibile filtrare i risultati utilizzando l'--include opzione --exclude o. L'--exclude opzione imposta le regole per escludere solo gli oggetti dal comando e le opzioni vengono applicate nell'ordine specificato. Questo viene mostrato nell'esempio seguente.

```
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"
```

## includere

Quando si utilizza il `s3 rm` comando `s3 cp`, `s3 mv`, `s3 sync`, o, è possibile filtrare i risultati utilizzando l'opzione `--include` o `--exclude`. L'opzione `--include` imposta le regole per includere solo gli oggetti specificati per il comando e le opzioni si applicano nell'ordine specificato. Questo viene mostrato nell'esempio seguente.

```
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"
```

## concessione

I comandi `s3 cp`, `s3 mv` e `s3 sync` includono un'opzione `--grants` che può essere utilizzata per concedere autorizzazioni sull'oggetto a utenti o gruppi specificati. Imposta l'opzione `--grants` su un elenco di autorizzazioni utilizzando la seguente sintassi. Sostituisci `Permission`, `Grantee_Type` e `Grantee_ID` con il tuo valore.

## Sintassi

```
--grants Permission=Grantee_Type=Grantee_ID  
        [Permission=Grantee_Type=Grantee_ID ...]
```

Ogni valore contiene i seguenti elementi:

- *Permission* — Specifica le autorizzazioni concesse. Può essere impostato su `read`, `readaclwriteacl`, o `full`
- *Grantee\_Type* : specifica come identificare il beneficiario. Può essere impostato su `uri`, `emailaddress` o `id`
- *Grantee\_ID* — specifica il beneficiario in base a *Grantee\_Type*.
  - `uri`— Quello del gruppo. URI Per ulteriori informazioni, vedi [Chi è un beneficiario?](#)
  - `emailaddress` - L'indirizzo e-mail dell'account.
  - `id` - L'ID canonico dell'account

Per ulteriori informazioni sul controllo degli accessi di Amazon S3, consulta [Controllo degli accessi](#).

L'esempio seguente copia un oggetto in un bucket. Concede `read` le autorizzazioni sull'oggetto a tutti e `full` le autorizzazioni (`readreadacl`, `ewriteacl`) all'account associato a `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
```

Puoi anche specificare una classe di storage non predefinita (`REDUCED_REDUNDANCY` o `STANDARD_IA`) per gli oggetti che carichi su Amazon S3. Per farlo, utilizza l'opzione `--storage-class`.

```
$ aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --storage-class REDUCED_REDUNDANCY
```

## recursive

Quando utilizzi questa opzione, il comando viene eseguito su tutti i file o gli oggetti nella directory o nel prefisso specificato. L'esempio seguente elimina tutto `s3://amzn-s3-demo-bucket/path` il relativo contenuto.



```
$ aws s3 rm s3://amzn-s3-demo-bucket/path --recursive
```

## Risorse

### AWS CLI riferimento:

- [aws s3](#)
- [aws s3 cp](#)
- [aws s3 mb](#)
- [aws s3 mv](#)
- [aws s3 ls](#)
- [aws s3 rb](#)
- [aws s3 rm](#)
- [aws s3 sync](#)

### Riferimento del servizio:

- [Utilizzo dei bucket Amazon S3](#) nella Guida per l'utente di Amazon S3
- [Utilizzo di oggetti Amazon S3](#) nella Amazon S3 User Guide
- [Elenco gerarchico delle chiavi utilizzando un prefisso e un delimitatore nella](#) Amazon S3 User Guide
- [Interrompi i caricamenti in più parti su un bucket S3 utilizzando la AWS SDK for .NET \(di basso livello\) nella Amazon S3 User Guide](#)

## Utilizzo dei API comandi -Level (s3api) in AWS CLI

I comandi API -level (contenuti nel set di s3api comandi) forniscono l'accesso diretto ad Amazon Simple Storage Service (Amazon S3) APIs e abilitano alcune operazioni che non sono esposte nei comandi di alto livello. s3 Questi comandi sono l'equivalente degli altri AWS servizi che forniscono accesso a API livello di accesso alle funzionalità dei servizi. Per ulteriori informazioni sui s3 comandi, vedere [Utilizzo di comandi di alto livello \(s3\) in AWS CLI](#)

Questo argomento fornisce esempi che dimostrano come utilizzare i comandi di livello inferiore mappati ad Amazon S3. APIs Inoltre, puoi trovare esempi per ogni API comando S3 nella s3api sezione della guida di riferimento 2 della [guida di AWS CLI riferimento](#).

## Argomenti

- [Prerequisiti](#)
- [Applica una personalizzazione ACL](#)
- [Configurazione di una policy di registrazione](#)
- [Risorse](#)

## Prerequisiti

Per eseguire i `s3api` comandi, devi:

- Installa e configura la AWS CLI. Per ulteriori informazioni, consulta [Installare il AWS CLI](#) e [Credenziali di autenticazione e accesso per AWS CLI](#).
- Il profilo utilizzato deve disporre di autorizzazioni che consentano le AWS operazioni eseguite dagli esempi.
- Comprendi questi termini di Amazon S3:
  - Bucket: cartella Amazon S3 di primo livello.
  - Prefisso: una cartella Amazon S3 in un bucket.
  - Oggetto: qualsiasi elemento ospitato in un bucket Amazon S3.

## Applica una personalizzazione ACL

Con i comandi di alto livello, puoi utilizzare l'`--acl` opzione per applicare elenchi di controllo degli accessi predefiniti (ACLs) agli oggetti Amazon S3. Ma non puoi usare quel comando per impostare bucket-wide ACLs. Tuttavia, puoi farlo usando il comando [put-bucket-acl](#) API -level.

L'esempio seguente mostra come concedere il controllo completo a due AWS utenti (`user1@example.com` e `user2@example.com`) e l'autorizzazione di lettura a tutti. L'identificatore per «tutti» proviene da uno speciale URI passato come parametro.

```
$ 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"'
```

Per dettagli su come costruire il ACLs, consulta [PUTBucket acl](#) nell'Amazon Simple Storage Service Reference. API I `s3api` ACL comandi in CLI, ad esempio `put-bucket-acl`, utilizzano la stessa notazione [abbreviata](#) degli argomenti.

## Configurazione di una policy di registrazione

Il API comando `put-bucket-logging` configura una politica di registrazione dei bucket.

Nell'esempio seguente, all' AWS utente `user@example.com` viene concesso il pieno controllo sui file di registro e tutti gli utenti hanno accesso in lettura ad essi. Nota che il `put-bucket-acl` comando è necessario anche per concedere al sistema di consegna dei log di Amazon S3 (specificato da `aURI`) le autorizzazioni necessarie per leggere e scrivere i log nel bucket.

```
$ 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
```

Il `logging.json` file del comando precedente ha il seguente contenuto.

```
{
  "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"
      }
    ]
  }
}
```

## Risorse

AWS CLI riferimento:

- [aws s3api](#)
- [aws s3api put-bucket-acl](#)
- [aws s3api put-bucket-logging](#)

Riferimento del servizio:

- [Utilizzo dei bucket Amazon S3](#) nella Guida per l'utente di Amazon S3
- [Utilizzo di oggetti Amazon S3](#) nella Amazon S3 User Guide
- [Elenco gerarchico delle chiavi utilizzando un prefisso e un delimitatore nella](#) Amazon S3 User Guide
- [Interrompi i caricamenti in più parti su un bucket S3 utilizzando la AWS SDK for .NET \(di basso livello\) nella Amazon S3 User Guide](#)

## Esempio di scripting per il ciclo di vita del bucket Amazon S3 nel AWS CLI

Questo argomento utilizza un esempio di script bash per le operazioni del ciclo di vita dei bucket Amazon S3 utilizzando (). AWS Command Line Interface AWS CLI Questo esempio di script utilizza il set di comandi. [aws s3api](#) Gli script di shell sono programmi progettati per essere eseguiti in un'interfaccia a riga di comando.

Argomenti

- [Prima di iniziare](#)
- [Informazioni su questo esempio](#)
- [File](#)
- [Riferimenti](#)

Prima di iniziare

Prima di poter eseguire uno qualsiasi degli esempi seguenti, è necessario completare le seguenti operazioni.

- Installa e configura la AWS CLI. Per ulteriori informazioni, consulta [Installare il AWS CLI](#) e [Credenziali di autenticazione e accesso per AWS CLI](#).
- Il profilo utilizzato deve disporre di autorizzazioni che consentano le AWS operazioni eseguite dagli esempi.

- Come procedura AWS consigliata, concedi a questo codice il privilegio minimo o solo le autorizzazioni necessarie per eseguire un'operazione. Per ulteriori informazioni, consulta [Grant Least Privilege nella Guida](#) per l'IAMutente.
- Questo codice non è stato testato in tutte le AWS regioni. Alcuni AWS servizi sono disponibili solo in regioni specifiche. Per ulteriori informazioni, consulta [Service Endpoints and Quotas](#) nella Guida di riferimento AWS generale.
- L'esecuzione di questo codice può comportare addebiti sul tuo AWS account. È tua responsabilità assicurarti che tutte le risorse create da questo script vengano rimosse quando hai finito di utilizzarle.

Il servizio Amazon S3 utilizza i seguenti termini:

- Bucket: una cartella Amazon S3 di primo livello.
- Prefisso: una cartella Amazon S3 in un bucket.
- Oggetto: qualsiasi elemento ospitato in un bucket Amazon S3.

Informazioni su questo esempio

Questo esempio dimostra come interagire con alcune delle operazioni di base di Amazon S3 utilizzando un set di funzioni nei file di script di shell. Le funzioni si trovano nel file di script di shell denominato `bucket-operations.sh`. È possibile richiamare queste funzioni in un altro file. Ogni file di script contiene commenti che descrivono ciascuna delle funzioni.

Per visualizzare i risultati intermedi di ogni passaggio, esegui lo script con un `-i` parametro. Puoi visualizzare lo stato corrente del bucket o il suo contenuto utilizzando la console Amazon S3. Lo script procede al passaggio successivo solo quando si preme invio al prompt.

Per l'esempio completo e i file di script scaricabili, consulta [Amazon S3 Bucket Lifecycle](#) Operations nel Code Examples Repository su AWS GitHub

File

L'esempio contiene i seguenti file:

`bucket-operations.sh`

Questo file di script principale può provenire da un altro file. Include funzioni che eseguono le seguenti attività:

- Creare un bucket e verificarne l'esistenza
- Copiare un file dal computer locale in un bucket
- Copiare un file da una posizione del bucket a un'altra posizione del bucket
- Elencare il contenuto di un bucket
- Eliminazione di un file da un bucket
- Eliminazione di un bucket

Visualizza il codice per [bucket-operations.sh](#) on GitHub

#### test-bucket-operations.sh

Il file di script di shell `test-bucket-operations.sh` mostra come chiamare le funzioni recuperando il `bucket-operations.sh` file e chiamando ciascuna delle funzioni. Dopo aver chiamato le funzioni, lo script di test rimuove tutte le risorse che ha creato.

Visualizza il codice per [test-bucket-operations.sh](#) on GitHub.

#### awsdocs-general.sh

Il file di script `awsdocs-general.sh` contiene funzioni generiche utilizzate negli esempi di codice avanzati per AWS CLI.

Visualizza il codice per [awsdocs-general.sh](#) on GitHub.

#### Riferimenti

##### AWS CLI riferimento:

- [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](#)

Altro riferimento:

- [Utilizzo dei bucket Amazon S3](#) nella Guida per l'utente di Amazon S3
- [Utilizzo di oggetti Amazon S3](#) nella Amazon S3 User Guide
- Per visualizzare e contribuire a AWS SDK creare esempi di AWS CLI codice, consulta il [AWS Code Examples Repository](#) su. GitHub

## Accesso ad Amazon SNS in AWS CLI

Puoi accedere alle funzionalità di Amazon Simple Notification Service (AmazonSNS) utilizzando AWS Command Line Interface (AWS CLI). Per elencare i AWS CLI comandi per AmazonSNS, usa il seguente comando.

```
aws sns help
```

Prima di eseguire qualsiasi comando, imposta le tue credenziali di default. Per ulteriori informazioni, consulta [Configurazione delle impostazioni per AWS CLI](#).

Questo argomento mostra esempi di AWS CLI comandi che eseguono attività comuni per AmazonSNS.

Argomenti

- [Creazione di un argomento](#)
- [Sottoscrizione a un argomento](#)
- [Pubblicazione in un argomento](#)
- [Annullamento della sottoscrizione a un argomento](#)
- [Eliminazione di un argomento](#)

### Creazione di un argomento

Per creare un argomento, utilizza il comando [sns create-topic](#) e specifica il nome da assegnare all'argomento.

```
$ aws sns create-topic --name my-topic
```

```
{
  "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
}
```

Prendi nota del valore `TopicArn` della risposta, che utilizzerai in seguito per pubblicare un messaggio.

## Sottoscrizione a un argomento

Per sottoscrivere un argomento, utilizza il comando [sns subscribe](#).

L'esempio seguente specifica il protocollo `email` e un indirizzo e-mail per `notification-endpoint`.

```
$ 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 invia immediatamente un messaggio di conferma via e-mail all'indirizzo specificato nel `subscribe` comando. Il testo dell'e-mail è il seguente.

```
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
```

Quando il destinatario fa clic sul collegamento `Confirm subscription` (Conferma sottoscrizione), il browser del destinatario visualizza un messaggio di notifica con informazioni simili alle seguenti.

```
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.
```



## Pubblicazione in un argomento

Per inviare un messaggio a tutti gli abbonati di un argomento, usa il [sns publish](#) comando.

L'esempio seguente invia il messaggio «Hello World!» a tutti gli abbonati dell'argomento specificato.

```
$ aws sns publish --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic --  
message "Hello World!"  
{  
  "MessageId": "4e41661d-5eec-5ddf-8dab-2c867EXAMPLE"  
}
```

In questo esempio, AWS invia un messaggio e-mail con il testo «Hello World!» a `asaanvi@example.com`.

## Annullamento della sottoscrizione a un argomento

Per annullare l'iscrizione a un argomento e interrompere la ricezione dei messaggi pubblicati su quell'argomento, usa il [sns unsubscribe](#) comando e specifica ARN l'argomento a cui desideri annullare l'iscrizione.

```
$ aws sns unsubscribe --subscription-arn arn:aws:sns:us-west-2:123456789012:my-  
topic:1328f057-de93-4c15-512e-8bb22EXAMPLE
```

Per verificare che l'iscrizione sia stata annullata correttamente, utilizzate il [sns list-subscriptions](#) comando per confermare che l'iscrizione ARN non sia più presente nell'elenco.

```
$ aws sns list-subscriptions
```

## Eliminazione di un argomento

Per eliminare un argomento, esegui il [sns delete-topic](#) comando.

```
$ aws sns delete-topic --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic
```

Per verificare che l'argomento sia stato eliminato AWS correttamente, utilizzate il [sns list-topics](#) comando per confermare che l'argomento non sia più presente nell'elenco.

```
$ aws sns list-topics
```

# AWS CLI esempi di comandi

Gli esempi di codice riportati in questo argomento mostrano come utilizzare AWS Command Line Interface with AWS.

Gli elementi di base sono esempi di codice che mostrano come eseguire le operazioni essenziali all'interno di un servizio.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto nei relativi scenari.

Gli scenari sono esempi di codice che mostrano come eseguire attività specifiche richiamando più funzioni all'interno di un servizio o combinandole con altre Servizi AWS.

## Servizi

- [ACMEsempi utilizzando AWS CLI](#)
- [APIEsempi di gateway che utilizzano AWS CLI](#)
- [APIGateway HTTP ed WebSocket API esempi di utilizzo AWS CLI](#)
- [APIAPIEsempi di gestione del gateway utilizzando AWS CLI](#)
- [Esempi di App Mesh utilizzando AWS CLI](#)
- [Esempi di App Runner utilizzando AWS CLI](#)
- [AWS AppConfig esempi utilizzando AWS CLI](#)
- [Esempi di Application Auto Scaling utilizzando AWS CLI](#)
- [Esempi di Application Discovery Service utilizzando AWS CLI](#)
- [AppRegistry esempi utilizzando AWS CLI](#)
- [Esempi di Athena che utilizzano AWS CLI](#)
- [Esempi di Auto Scaling utilizzando AWS CLI](#)
- [Esempi di Piani di Auto Scaling che utilizzano AWS CLI](#)
- [AWS Backup esempi utilizzando AWS CLI](#)
- [AWS Batch esempi utilizzando AWS CLI](#)
- [Budget AWS esempi utilizzando AWS CLI](#)
- [Esempi di utilizzo di Amazon Chime AWS CLI](#)
- [APIEsempi di Cloud Control utilizzando AWS CLI](#)

- [AWS Cloud Map esempi utilizzando AWS CLI](#)
- [AWS Cloud9 esempi utilizzando AWS CLI](#)
- [AWS CloudFormation esempi utilizzando AWS CLI](#)
- [CloudFront esempi utilizzando AWS CLI](#)
- [CloudSearch Esempi di utilizzo di Amazon AWS CLI](#)
- [CloudTrail esempi utilizzando AWS CLI](#)
- [CloudWatch esempi utilizzando AWS CLI](#)
- [CloudWatch Esempi di log utilizzando AWS CLI](#)
- [CloudWatch Esempi di monitoraggio della rete utilizzando AWS CLI](#)
- [CodeArtifact esempi utilizzando AWS CLI](#)
- [CodeBuild esempi utilizzando AWS CLI](#)
- [CodeCommit esempi utilizzando AWS CLI](#)
- [CodeDeploy esempi utilizzando AWS CLI](#)
- [CodeGuru Esempi di revisori che utilizzano AWS CLI](#)
- [CodePipeline esempi utilizzando AWS CLI](#)
- [AWS CodeStar Esempi di notifiche utilizzando AWS CLI](#)
- [CodeConnections esempi utilizzando AWS CLI](#)
- [Esempi di identità di Amazon Cognito utilizzando AWS CLI](#)
- [Esempi di Amazon Cognito Identity Provider che utilizzano AWS CLI](#)
- [Esempi di Amazon Comprehend con AWS CLI](#)
- [Esempi di utilizzo di Amazon Comprehend Medical AWS CLI](#)
- [AWS Config esempi utilizzando AWS CLI](#)
- [Esempi di utilizzo di Amazon Connect AWS CLI](#)
- [AWS Cost and Usage Report esempi utilizzando AWS CLI](#)
- [Esempi di utilizzo del servizio Cost Explorer AWS CLI](#)
- [Esempi di utilizzo di Firehose AWS CLI](#)
- [Esempi di utilizzo di Amazon Data Lifecycle Manager AWS CLI](#)
- [AWS Data Pipeline esempi utilizzando AWS CLI](#)
- [DataSync esempi utilizzando AWS CLI](#)
- [DAXesempi utilizzando AWS CLI](#)

- [Esempi di Detective che utilizzano AWS CLI](#)
- [Esempi di Device Farm utilizzando AWS CLI](#)
- [AWS Direct Connect esempi utilizzando AWS CLI](#)
- [AWS Directory Service esempi utilizzando AWS CLI](#)
- [AWS DMS esempi utilizzando AWS CLI](#)
- [Esempi di utilizzo di Amazon DocumentDB AWS CLI](#)
- [Esempi di utilizzo di DynamoDB AWS CLI](#)
- [Esempi di DynamoDB Streams che utilizzano AWS CLI](#)
- [EC2Esempi di utilizzo di Amazon AWS CLI](#)
- [Esempi di Amazon EC2 Instance Connect con AWS CLI](#)
- [ECREsempi di utilizzo di Amazon AWS CLI](#)
- [Esempi di Amazon ECR Public che utilizzano AWS CLI](#)
- [ECSEsempi di utilizzo di Amazon AWS CLI](#)
- [EFSEsempi di utilizzo di Amazon AWS CLI](#)
- [EKSEsempi di utilizzo di Amazon AWS CLI](#)
- [Esempi di utilizzo di Elastic Beanstalk AWS CLI](#)
- [Elastic Load Balancing - Esempi di utilizzo della versione 1 AWS CLI](#)
- [Elastic Load Balancing - Esempi di utilizzo della versione 2 AWS CLI](#)
- [Esempi di utilizzo di Elastic Transcoder AWS CLI](#)
- [ElastiCache esempi utilizzando AWS CLI](#)
- [MediaStore esempi utilizzando AWS CLI](#)
- [EMREsempi di utilizzo di Amazon AWS CLI](#)
- [Amazon EMR su EKS esempi di utilizzo AWS CLI](#)
- [EventBridge esempi utilizzando AWS CLI](#)
- [Esempi di Firewall Manager che utilizzano AWS CLI](#)
- [AWS FIS esempi utilizzando AWS CLI](#)
- [GameLift Esempi di utilizzo di Amazon AWS CLI](#)
- [Esempi di Global Accelerator che utilizzano AWS CLI](#)
- [AWS Glue esempi utilizzando AWS CLI](#)
- [GuardDuty esempi utilizzando AWS CLI](#)

- [AWS Health esempi utilizzando AWS CLI](#)
- [HealthImaging esempi utilizzando AWS CLI](#)
- [HealthLake esempi utilizzando AWS CLI](#)
- [HealthOmics esempi utilizzando AWS CLI](#)
- [IAMesempi utilizzando AWS CLI](#)
- [IAMEsempi di Access Analyzer utilizzando AWS CLI](#)
- [Esempi di Image Builder che utilizzano AWS CLI](#)
- [Esempi di Incident Manager che utilizzano AWS CLI](#)
- [Esempi di contatti di Incident Manager utilizzando AWS CLI](#)
- [Esempi di utilizzo di Amazon Inspector AWS CLI](#)
- [AWS IoT esempi utilizzando AWS CLI](#)
- [AWS IoT 1-Click Esempi di dispositivi che utilizzano AWS CLI](#)
- [AWS IoT 1-Click Esempi di progetti che utilizzano AWS CLI](#)
- [AWS IoT Analytics esempi utilizzando AWS CLI](#)
- [Esempi di Device Advisor utilizzando AWS CLI](#)
- [AWS IoT data esempi utilizzando AWS CLI](#)
- [AWS IoT Events esempi utilizzando AWS CLI](#)
- [AWS IoT Events-Data esempi utilizzando AWS CLI](#)
- [AWS IoT Greengrass esempi utilizzando AWS CLI](#)
- [AWS IoT Greengrass V2 esempi utilizzando AWS CLI](#)
- [AWS IoT Jobs SDK release esempi utilizzando AWS CLI](#)
- [AWS IoT SiteWise esempi utilizzando AWS CLI](#)
- [AWS IoT Things Graph esempi utilizzando AWS CLI](#)
- [Wireless AWS IoT esempi utilizzando AWS CLI](#)
- [IVSEsempi di utilizzo di Amazon AWS CLI](#)
- [Esempi di Amazon IVS Chat utilizzando AWS CLI](#)
- [Esempi di Amazon IVS Real-Time Streaming utilizzando AWS CLI](#)
- [Esempi di utilizzo di Amazon Kendra AWS CLI](#)
- [Esempi di Kinesis che utilizzano AWS CLI](#)
- [AWS KMS esempi utilizzando AWS CLI](#)

- [Esempi di Lake Formation utilizzando AWS CLI](#)
- [Esempi di utilizzo di Lambda AWS CLI](#)
- [Esempi di License Manager che utilizzano AWS CLI](#)
- [Esempi di utilizzo di Lightsail AWS CLI](#)
- [Esempi di Macie che utilizzano AWS CLI](#)
- [Esempi di Amazon Managed Grafana utilizzando AWS CLI](#)
- [MediaConnect esempi utilizzando AWS CLI](#)
- [MediaConvert esempi utilizzando AWS CLI](#)
- [MediaLive esempi utilizzando AWS CLI](#)
- [MediaPackage esempi utilizzando AWS CLI](#)
- [MediaPackage VOD esempi utilizzando AWS CLI](#)
- [MediaStore Esempi di piani dati che utilizzano AWS CLI](#)
- [MediaTailor esempi utilizzando AWS CLI](#)
- [Esempi di MemoryDB utilizzando AWS CLI](#)
- [MSK Esempi di utilizzo di Amazon AWS CLI](#)
- [Esempi di Network Manager che utilizzano AWS CLI](#)
- [Esempi di utilizzo di Nimble Studio AWS CLI](#)
- [OpenSearch Esempi di servizi che utilizzano AWS CLI](#)
- [AWS OpsWorks esempi utilizzando AWS CLI](#)
- [AWS OpsWorks CM esempi utilizzando AWS CLI](#)
- [Organizations: esempi che utilizzano AWS CLI](#)
- [AWS Outposts esempi utilizzando AWS CLI](#)
- [AWS Payment Cryptography esempi utilizzando AWS CLI](#)
- [AWS Payment Cryptography Esempi di piani dati che utilizzano AWS CLI](#)
- [Esempi di utilizzo di Amazon Pinpoint AWS CLI](#)
- [Esempi di utilizzo di Amazon Polly AWS CLI](#)
- [Listino prezzi AWS esempi utilizzando AWS CLI](#)
- [AWS Private CA esempi utilizzando AWS CLI](#)
- [AWS Proton esempi utilizzando AWS CLI](#)
- [QLDB esempi utilizzando AWS CLI](#)

- [RDSEsempi di utilizzo di Amazon AWS CLI](#)
- [Esempi di utilizzo di Amazon RDS Data Service AWS CLI](#)
- [Esempi di utilizzo di Amazon RDS Performance Insights AWS CLI](#)
- [Esempi di utilizzo di Amazon Redshift AWS CLI](#)
- [Esempi di utilizzo di Amazon Rekognition AWS CLI](#)
- [AWS RAM esempi utilizzando AWS CLI](#)
- [Esempi di Resource Explorer utilizzando AWS CLI](#)
- [Esempi di Resource Groups utilizzando AWS CLI](#)
- [API Esempi di etichettatura dei Resource Groups utilizzando AWS CLI](#)
- [AWS RoboMaker esempi utilizzando AWS CLI](#)
- [Esempi di Route 53 che utilizzano AWS CLI](#)
- [Esempi di registrazione del dominio Route 53 utilizzando AWS CLI](#)
- [Esempi di profili Route 53 utilizzando AWS CLI](#)
- [Esempi di Route 53 Resolver che utilizzano AWS CLI](#)
- [Esempi di utilizzo di Amazon S3 AWS CLI](#)
- [Esempi di Amazon S3 Control con AWS CLI](#)
- [Esempi di utilizzo di S3 Glacier AWS CLI](#)
- [Esempi di Secrets Manager che utilizzano AWS CLI](#)
- [Esempi di Security Hub che utilizzano AWS CLI](#)
- [Esempi di utilizzo di Security Lake AWS CLI](#)
- [AWS Serverless Application Repository esempi utilizzando AWS CLI](#)
- [Esempi di Service Catalog utilizzando AWS CLI](#)
- [Esempi di Service Quotas utilizzando AWS CLI](#)
- [SESEsempi di utilizzo di Amazon AWS CLI](#)
- [Esempi di utilizzo di Shield AWS CLI](#)
- [Esempi di firmatari che utilizzano AWS CLI](#)
- [Esempi di utilizzo di Snowball AWS CLI](#)
- [SNSEsempi di utilizzo di Amazon AWS CLI](#)
- [SQSEsempi di utilizzo di Amazon AWS CLI](#)
- [Esempi di Storage Gateway utilizzando AWS CLI](#)

- [AWS STS esempi utilizzando AWS CLI](#)
- [AWS Support esempi utilizzando AWS CLI](#)
- [SWFEsempi di utilizzo di Amazon AWS CLI](#)
- [Esempi di Systems Manager che utilizzano AWS CLI](#)
- [Esempi di utilizzo di Amazon Textract AWS CLI](#)
- [Esempi di Amazon Transcribe utilizzando AWS CLI](#)
- [Esempi di Amazon Translate con AWS CLI](#)
- [Trusted Advisor esempi utilizzando AWS CLI](#)
- [Esempi di autorizzazioni verificate utilizzando AWS CLI](#)
- [VPCEsempi di reticoli utilizzando AWS CLI](#)
- [AWS WAF Classic esempi utilizzando AWS CLI](#)
- [AWS WAF Classic regionale esempi utilizzando AWS CLI](#)
- [AWS WAFV2 esempi utilizzando AWS CLI](#)
- [WorkDocs Esempi di utilizzo di Amazon AWS CLI](#)
- [WorkMail Esempi di utilizzo di Amazon AWS CLI](#)
- [Esempi di Amazon WorkMail Message Flow con AWS CLI](#)
- [WorkSpaces esempi utilizzando AWS CLI](#)
- [Esempi di raggi X utilizzando AWS CLI](#)

## ACMe esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with ACM.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)



## Azioni

### **add-tags-to-certificate**

Il seguente esempio di codice mostra come utilizzare `add-tags-to-certificate`.

#### AWS CLI

Per aggiungere tag a un ACM certificato esistente

Il `add-tags-to-certificate` comando seguente aggiunge due tag al certificato specificato. Usa uno spazio per separare più tag:

```
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
```

- Per API i dettagli, vedere [AddTagsToCertificate](#) in AWS CLI Command Reference.

### **delete-certificate**

Il seguente esempio di codice mostra come utilizzare `delete-certificate`.

#### AWS CLI

Per eliminare un ACM certificato dal tuo account

Il `delete-certificate` comando seguente elimina il certificato con il valore specificato ARN:

```
aws acm delete-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

- Per API i dettagli, vedere [DeleteCertificate](#) in AWS CLI Command Reference.

### **describe-certificate**

Il seguente esempio di codice mostra come utilizzare `describe-certificate`.

#### AWS CLI

Per recuperare i campi contenuti in un certificato ACM

Il `describe-certificate` comando seguente recupera tutti i campi del certificato con i dati specificati: ARN

```
aws acm describe-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

Viene visualizzato un output simile al seguente:

```
{
  "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"
    ]
  }
}

```

- Per API i dettagli, vedere [DescribeCertificate](#) in AWS CLI Command Reference.

## export-certificate

Il seguente esempio di codice mostra come utilizzare `export-certificate`.

### AWS CLI

Per esportare un certificato privato emesso da una CA privata.

Il `export-certificate` comando seguente esporta un certificato privato, una catena di certificati e una chiave privata sul display:

```

aws acm export-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
passphrase file://path-to-passphrase-file

```

Per esportare il certificato, la catena e la chiave privata in un file locale, utilizzate il seguente comando:

```

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

```

- Per API i dettagli, vedere [ExportCertificate](#) in AWS CLI Command Reference.

## get-certificate

Il seguente esempio di codice mostra come utilizzare `get-certificate`.

### AWS CLI

Per recuperare un certificato ACM

Il `get-certificate` comando seguente recupera il certificato per il certificato specificato ARN e per la catena di certificati:

```
aws acm get-certificate --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

Viene visualizzato un output simile al seguente:

```
{
  "Certificate": "-----BEGIN CERTIFICATE-----
MIICiTCCAFICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLygVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
  "CertificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCCAFICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
```

```

b2x1MRIwEAYDVQQDEw1UZsXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVvxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
"-----BEGIN CERTIFICATE-----
MIICiTCcAfICcQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxGzAxBG9NVBAgTAldBMRAdDgYDVQQHEwZDZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xZDASBgNVBA5TC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZsXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxGzAxBG9NVBAgTAldBMRAdDgYD
VQQHEwZDZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xZDASBgNVBA5TC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZsXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVvxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
"-----BEGIN CERTIFICATE-----
MIICiTCcAfICcQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxGzAxBG9NVBAgTAldBMRAdDgYDVQQHEwZDZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xZDASBgNVBA5TC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZsXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxGzAxBG9NVBAgTAldBMRAdDgYD
VQQHEwZDZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xZDASBgNVBA5TC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZsXN0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVvxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----"
}

```

- Per API i dettagli, vedere [GetCertificate](#) in AWS CLI Command Reference.

## import-certificate

Il seguente esempio di codice mostra come utilizzare `import-certificate`.

### AWS CLI

Per importare un certificato in ACM.

Il `import-certificate` comando seguente importa un certificato in ACM. Sostituisci i nomi dei file con i tuoi:

```
aws acm import-certificate --certificate file://Certificate.pem --certificate-chain file://CertificateChain.pem --private-key file://PrivateKey.pem
```

- Per API i dettagli, vedere [ImportCertificate](#) in AWS CLI Command Reference.

## list-certificates

Il seguente esempio di codice mostra come utilizzare `list-certificates`.

### AWS CLI

Per elencare i ACM certificati di un AWS account

Il `list-certificates` comando seguente elenca ARNs i certificati presenti nell'account:

```
aws acm list-certificates
```

Il comando precedente genera un output simile al seguente:

```
{
  "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"
    }
  ]
}
```

```

    }
  ]
}

```

Puoi decidere quanti certificati vuoi visualizzare ogni volta che chiami `list-certificates`. Ad esempio, se avete quattro certificati e non volete visualizzarne più di due alla volta, impostate l'`max-items` argomento su 2 come nell'esempio seguente:

```
aws acm list-certificates --max-items 2
```

Verranno visualizzati due certificati ARNs e un `NextToken` valore:

```

"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"

```

Per visualizzare i prossimi due certificati nel tuo account, imposta questo `NextToken` valore nella prossima chiamata:

```
aws acm list-certificates --max-items 2 --next-token 9f4d9f69-275a-41fe-  
b58e-2b837bd9ba48
```

È possibile filtrare l'output utilizzando l'`certificate-statuses` argomento. Il comando seguente visualizza i certificati con `VALIDATION` lo stato `PENDING` \_:

```
aws acm list-certificates --certificate-statuses PENDING_VALIDATION
```

È inoltre possibile filtrare l'output utilizzando l'`includes` argomento. Il comando seguente visualizza i certificati filtrati in base alle seguenti proprietà. I certificati da visualizzare:

- 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
```

- Per API i dettagli, vedere [ListCertificates](#) in AWS CLI Command Reference.

## list-tags-for-certificate

Il seguente esempio di codice mostra come utilizzare `list-tags-for-certificate`.

### AWS CLI

Per elencare i tag applicati a un ACM certificato

Il `list-tags-for-certificate` comando seguente elenca i tag applicati a un certificato nel tuo account:

```
aws acm list-tags-for-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

Il comando precedente genera un output simile al seguente:

```
{
  "Tags": [
    {
      "Value": "Website",
      "Key": "Purpose"
    },
    {
      "Value": "Alice",
      "Key": "Admin"
    }
  ]
}
```

- Per API i dettagli, vedere [ListTagsForCertificate](#) in AWS CLI Command Reference.



## remove-tags-from-certificate

Il seguente esempio di codice mostra come utilizzare `remove-tags-from-certificate`.

### AWS CLI

Per rimuovere un tag da un ACM certificato

Il `remove-tags-from-certificate` comando seguente rimuove due tag dal certificato specificato. Usa uno spazio per separare più tag:

```
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
```

- Per API i dettagli, vedere [RemoveTagsFromCertificate](#) in AWS CLI Command Reference.

## request-certificate

Il seguente esempio di codice mostra come utilizzare `request-certificate`.

### AWS CLI

Per richiedere un nuovo ACM certificato

Il `request-certificate` comando seguente richiede un nuovo certificato per il dominio `www.example.com` utilizzando la convalida: DNS

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS
```

È possibile inserire un token di idempotenza per distinguere le chiamate a: `request-certificate`

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --idempotency-token 91adc45q
```

Puoi inserire uno o più nomi alternativi di soggetto per richiedere un certificato che protegga più di un dominio apex:

```
aws acm request-certificate --domain-name example.com --validation-method DNS --idempotency-token 91adc45q --subject-alternative-names www.example.net
```

Puoi inserire un nome alternativo che può essere utilizzato anche per raggiungere il tuo sito web:

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
idempotency-token 91adc45q --subject-alternative-names www.example.com
```

Puoi usare un asterisco (\*) come jolly per creare un certificato per diversi sottodomini dello stesso dominio:

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
idempotency-token 91adc45q --subject-alternative-names *.example.com
```

Puoi anche inserire più nomi alternativi:

```
aws acm request-certificate --domain-name example.com --validation-method DNS --  
subject-alternative-names b.example.com c.example.com d.example.com
```

Se utilizzi l'e-mail per la convalida, puoi inserire le opzioni di convalida del dominio per specificare il dominio a cui verrà inviata l'e-mail di convalida:

```
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
```

Il comando seguente disattiva la registrazione della trasparenza dei certificati quando richiedi un nuovo certificato:

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --  
options CertificateTransparencyLoggingPreference=DISABLED --idempotency-token 184627
```

- Per API i dettagli, vedere [RequestCertificate](#) in AWS CLI Command Reference.

## resend-validation-email

Il seguente esempio di codice mostra come utilizzare `resend-validation-email`.

### AWS CLI

Per inviare nuovamente l'e-mail di convalida per la richiesta ACM di certificato

Il `resend-validation-email` comando seguente indica all'autorità di certificazione Amazon di inviare e-mail di convalida agli indirizzi appropriati:

```
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
```

- Per API i dettagli, consulta [ResendValidationEmail AWS CLI Command Reference](#).

## update-certificate-options

Il seguente esempio di codice mostra come utilizzare `update-certificate-options`.

### AWS CLI

Per aggiornare le opzioni del certificato

Il `update-certificate-options` comando seguente disattiva la registrazione della trasparenza dei certificati:

```
aws acm update-certificate-options --certificate-arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --options CertificateTransparencyLoggingPreference=DISABLED
```

- Per API i dettagli, vedere [UpdateCertificateOptions](#) in AWS CLI Command Reference.

## API Esempi di gateway che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with API Gateway.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **create-api-key**

Il seguente esempio di codice mostra come utilizzare `create-api-key`.

#### AWS CLI

Per creare una API chiave abilitata per uno Stage esistente API

Comando:

```
aws apigateway create-api-key --name 'Dev API Key' --description 'Used for development' --enabled --stage-keys restApiId='a1b2c3d4e5',stageName='dev'
```

- Per API i dettagli, vedere [CreateApiKey](#) in AWS CLI Command Reference.

### **create-authorizer**

Il seguente esempio di codice mostra come utilizzare `create-authorizer`.

#### AWS CLI

Esempio 1: creare un API Gateway Custom Authorizer basato su token per API

L'`create-authorizer` esempio seguente crea un autorizzatore basato su token.

```
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
```

Output:

```
{  
  "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"
}

```

Esempio 2: creare un API Gateway Custom Authorizer basato su Cognito User Pools per API

L'create-authorizeresempio seguente crea un API Gateway Custom Authorizer basato su Cognito User Pools.

```

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'

```

Output:

```

{
  "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"
}

```

Esempio 3: creare un API Gateway Custom Authorizer basato su richiesta per API

L'create-authorizeresempio seguente crea un autorizzatore basato su richiesta.

```

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
```

Output:

```
{
  "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
}
```

- Per API i dettagli, vedere [CreateAuthorizer](#) in AWS CLI Command Reference.

## create-base-path-mapping

Il seguente esempio di codice mostra come utilizzare `create-base-path-mapping`.

AWS CLI

Per creare la mappatura del percorso di base per un nome di dominio personalizzato

Comando:

```
aws apigateway create-base-path-mapping --domain-name subdomain.domain.tld --rest-
api-id 1234123412 --stage prod --base-path v1
```

- Per API i dettagli, vedere [CreateBasePathMapping](#) in AWS CLI Command Reference.

## create-deployment

Il seguente esempio di codice mostra come utilizzare `create-deployment`.

AWS CLI

Per distribuire le risorse configurate per An API in un nuovo Stage

Comando:

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --stage-description 'Development Stage' --description 'First deployment to the dev stage'
```

Per distribuire le risorse configurate API per una fase esistente

Comando:

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --description 'Second deployment to the dev stage'
```

Per distribuire le risorse configurate per un API su una fase esistente con Stage Variables

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --description 'Terza implementazione alla fase di sviluppo' --variables key='value', = 'otherKeyotherValue'
```

- Per API i dettagli, AWS CLI vedi in Command Reference. [CreateDeployment](#)

## create-domain-name

Il seguente esempio di codice mostra come utilizzare `create-domain-name`.

AWS CLI

Per creare il nome di dominio personalizzato

Comando:

```
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'
```

- Per API i dettagli, vedere [CreateDomainName](#) in AWS CLI Command Reference.

## create-model

Il seguente esempio di codice mostra come utilizzare `create-model`.

AWS CLI

Per creare un modello per un API

Comando:

```
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" } } } } }'
```

Output:

```
{
  "contentType": "application/json",
  "description": "The First Model",
  "name": "firstModel",
  "id": "2rzg0l",
  "schema": "{ \"$schema\": \"http://json-schema.org/draft-04/schema#\", \"title\": \"firstModel\", \"type\": \"object\", \"properties\": { \"firstProperty\": { \"type\": \"object\", \"properties\": { \"key\": { \"type\": \"string\" } } } } }"
```

- Per API i dettagli, vedere [CreateModel](#) in AWS CLI Command Reference.

## create-resource

Il seguente esempio di codice mostra come utilizzare `create-resource`.

AWS CLI

Per creare una risorsa in un API

Comando:

```
aws apigateway create-resource --rest-api-id 1234123412 --parent-id a1b2c3 --path-part 'new-resource'
```

- Per API i dettagli, vedere [CreateResource](#) in AWS CLI Command Reference.

## create-rest-api

Il seguente esempio di codice mostra come utilizzare `create-rest-api`.



## AWS CLI

Per creare un API

Comando:

```
aws apigateway create-rest-api --name 'My First API' --description 'This is my first API'
```

Per creare un duplicato API da un file esistente API

Comando:

```
aws apigateway create-rest-api --name 'Copy of My First API' --description 'This is a copy of my first API' --clone-from 1234123412
```

- Per API i dettagli, vedere [CreateRestApi](#) in AWS CLI Command Reference.

## create-stage

Il seguente esempio di codice mostra come utilizzare `create-stage`.

## AWS CLI

Creare una fase in un ambiente API che contenga una distribuzione esistente

Comando:

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev' --description 'Development stage' --deployment-id a1b2c3
```

Per creare una fase in un ambiente API che conterrà una distribuzione esistente e variabili di fase personalizzate

Comando:

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev' --description 'Development stage' --deployment-id a1b2c3 --variables key='value',otherKey='otherValue'
```

- Per API i dettagli, vedere [CreateStage](#) in AWS CLI Command Reference.

## create-usage-plan-key

Il seguente esempio di codice mostra come utilizzare `create-usage-plan-key`.

### AWS CLI

Associa una API chiave esistente a un piano di utilizzo

Comando:

```
aws apigateway create-usage-plan-key --usage-plan-id a1b2c3 --key-type "API_KEY" --key-id 4vq3yryqm5
```

- Per API i dettagli, vedere [CreateUsagePlanKey](#) in AWS CLI Command Reference.

## create-usage-plan

Il seguente esempio di codice mostra come utilizzare `create-usage-plan`.

### AWS CLI

Per creare un piano di utilizzo con limiti di limitazione e quota che si ripristini all'inizio del mese

Comando:

```
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
```

- Per API i dettagli, consulta Command [CreateUsagePlan](#) Reference AWS CLI .

## delete-api-key

Il seguente esempio di codice mostra come utilizzare `delete-api-key`.

### AWS CLI

Per eliminare una API chiave

Comando:

```
aws apigateway delete-api-key --api-key 8bk1k8b11k3sB38D9B310enyWT8c09B301kq0b1k
```

- Per API i dettagli, vedere [DeleteApiKey](#) in AWS CLI Command Reference.

## **delete-authorizer**

Il seguente esempio di codice mostra come utilizzare `delete-authorizer`.

AWS CLI

Per eliminare un Custom Authorizer in un API

Comando:

```
aws apigateway delete-authorizer --rest-api-id 1234123412 --authorizer-id 7gkfbo
```

- Per API i dettagli, vedere [DeleteAuthorizer](#) in AWS CLI Command Reference.

## **delete-base-path-mapping**

Il seguente esempio di codice mostra come utilizzare `delete-base-path-mapping`.

AWS CLI

Per eliminare una mappatura del percorso di base per un nome di dominio personalizzato

Comando:

```
aws apigateway delete-base-path-mapping --domain-name 'api.domain.tld' --base-path 'dev'
```

- Per API i dettagli, vedere [DeleteBasePathMapping](#) in AWS CLI Command Reference.

## **delete-client-certificate**

Il seguente esempio di codice mostra come utilizzare `delete-client-certificate`.

## AWS CLI

Per eliminare un certificato client

Comando:

```
aws apigateway delete-client-certificate --client-certificate-id a1b2c3
```

- Per API i dettagli, vedere [DeleteClientCertificate](#) in AWS CLI Command Reference.

## **delete-deployment**

Il seguente esempio di codice mostra come utilizzare `delete-deployment`.

### AWS CLI

Per eliminare una distribuzione in un API

Comando:

```
aws apigateway delete-deployment --rest-api-id 1234123412 --deployment-id a1b2c3
```

- Per API i dettagli, vedere [DeleteDeployment](#) in AWS CLI Command Reference.

## **delete-domain-name**

Il seguente esempio di codice mostra come utilizzare `delete-domain-name`.

### AWS CLI

Per eliminare un nome di dominio personalizzato

Comando:

```
aws apigateway delete-domain-name --domain-name 'api.domain.tld'
```

- Per API i dettagli, vedere [DeleteDomainName](#) in AWS CLI Command Reference.

## **delete-integration-response**

Il seguente esempio di codice mostra come utilizzare `delete-integration-response`.

## AWS CLI

Per eliminare una risposta di integrazione per una determinata risorsa, metodo e codice di stato in un API

Comando:

```
aws apigateway delete-integration-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 200
```

- Per API i dettagli, vedere [DeleteIntegrationResponse](#) in AWS CLI Command Reference.

### **delete-integration**

Il seguente esempio di codice mostra come utilizzare `delete-integration`.

## AWS CLI

Per eliminare un'integrazione per una determinata risorsa e metodo in un API

Comando:

```
aws apigateway delete-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET
```

- Per API i dettagli, vedere [DeleteIntegration](#) in AWS CLI Command Reference.

### **delete-method-response**

Il seguente esempio di codice mostra come utilizzare `delete-method-response`.

## AWS CLI

Per eliminare una risposta al metodo per la risorsa, il metodo e il codice di stato specificati in un API

Comando:

```
aws apigateway delete-method-response --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --status-code 200
```

- Per API i dettagli, vedere [DeleteMethodResponse](#) in AWS CLI Command Reference.

## delete-method

Il seguente esempio di codice mostra come utilizzare `delete-method`.

### AWS CLI

Per eliminare un metodo per una determinata risorsa in un API

Comando:

```
aws apigateway delete-method --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET
```

- Per API i dettagli, vedere [DeleteMethod](#) in AWS CLI Command Reference.

## delete-model

Il seguente esempio di codice mostra come utilizzare `delete-model`.

### AWS CLI

Eliminare un modello in un dato API

Comando:

```
aws apigateway delete-model --rest-api-id 1234123412 --model-name 'customModel'
```

- Per API i dettagli, vedere [DeleteModel](#) in AWS CLI Command Reference.

## delete-resource

Il seguente esempio di codice mostra come utilizzare `delete-resource`.

### AWS CLI

Per eliminare una risorsa in un API

Comando:

```
aws apigateway delete-resource --rest-api-id 1234123412 --resource-id a1b2c3
```

- Per API i dettagli, vedere [DeleteResource](#) in AWS CLI Command Reference.

## delete-rest-api

Il seguente esempio di codice mostra come utilizzare `delete-rest-api`.

### AWS CLI

Per eliminare un API

Comando:

```
aws apigateway delete-rest-api --rest-api-id 1234123412
```

- Per API i dettagli, vedere [DeleteRestApi](#) in AWS CLI Command Reference.

## delete-stage

Il seguente esempio di codice mostra come utilizzare `delete-stage`.

### AWS CLI

Per eliminare una fase in un API

Comando:

```
aws apigateway delete-stage --rest-api-id 1234123412 --stage-name 'dev'
```

- Per API i dettagli, vedere [DeleteStage](#) in AWS CLI Command Reference.

## delete-usage-plan-key

Il seguente esempio di codice mostra come utilizzare `delete-usage-plan-key`.

### AWS CLI

Per rimuovere una API chiave da un piano di utilizzo

Comando:

```
aws apigateway delete-usage-plan-key --usage-plan-id a1b2c3 --key-id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- Per API i dettagli, vedere [DeleteUsagePlanKey](#) in AWS CLI Command Reference.

## delete-usage-plan

Il seguente esempio di codice mostra come utilizzare `delete-usage-plan`.

### AWS CLI

Per eliminare un piano di utilizzo

Comando:

```
aws apigateway delete-usage-plan --usage-plan-id a1b2c3
```

- Per API i dettagli, vedere [DeleteUsagePlan](#) in AWS CLI Command Reference.

## flush-stage-authorizers-cache

Il seguente esempio di codice mostra come utilizzare `flush-stage-authorizers-cache`.

### AWS CLI

Per svuotare tutte le voci della cache dell'autorizzatore su uno stage

Comando:

```
aws apigateway flush-stage-authorizers-cache --rest-api-id 1234123412 --stage-name dev
```

- Per API i dettagli, vedere [FlushStageAuthorizersCache](#) in AWS CLI Command Reference.

## flush-stage-cache

Il seguente esempio di codice mostra come utilizzare `flush-stage-cache`.

### AWS CLI

Per svuotare la cache per uno API stage



Comando:

```
aws apigateway flush-stage-cache --rest-api-id 1234123412 --stage-name dev
```

- Per API i dettagli, vedere [FlushStageCache](#) in AWS CLI Command Reference.

## generate-client-certificate

Il seguente esempio di codice mostra come utilizzare `generate-client-certificate`.

AWS CLI

Per creare un certificato lato client SSL

Comando:

```
aws apigateway generate-client-certificate --description 'My First Client Certificate'
```

- Per API i dettagli, vedere [GenerateClientCertificate](#) in AWS CLI Command Reference.

## get-account

Il seguente esempio di codice mostra come utilizzare `get-account`.

AWS CLI

Per ottenere le impostazioni dell'account API Gateway

Comando:

```
aws apigateway get-account
```

Output:

```
{
  "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/APIGatewayToCloudWatchLogsRole",
  "throttleSettings": {
    "rateLimit": 500.0,
    "burstLimit": 1000
  }
}
```

```
}  
}
```

- Per API i dettagli, vedere [GetAccount](#) in AWS CLI Command Reference.

## get-api-key

Il seguente esempio di codice mostra come utilizzare `get-api-key`.

### AWS CLI

Per ottenere le informazioni su una API chiave specifica

Comando:

```
aws apigateway get-api-key --api-key 8bk1k8b11k3sB38D9B310enyWT8c09B301kq0b1k
```

Output:

```
{  
  "description": "My first key",  
  "enabled": true,  
  "stageKeys": [  
    "a1b2c3d4e5/dev",  
    "e5d4c3b2a1/dev"  
  ],  
  "lastUpdatedDate": 1456184515,  
  "createdDate": 1456184452,  
  "id": "8bk1k8b11k3sB38D9B310enyWT8c09B301kq0b1k",  
  "name": "My key"  
}
```

- Per API i dettagli, vedere [GetApiKey](#) in AWS CLI Command Reference.

## get-api-keys

Il seguente esempio di codice mostra come utilizzare `get-api-keys`.

### AWS CLI

Per ottenere l'elenco delle API chiavi

Comando:

```
aws apigateway get-api-keys
```

Output:

```
{
  "items": [
    {
      "description": "My first key",
      "enabled": true,
      "stageKeys": [
        "a1b2c3d4e5/dev",
        "e5d4c3b2a1/dev"
      ],
      "lastUpdatedDate": 1456184515,
      "createdDate": 1456184452,
      "id": "8bk1k8b11k3sB38D9B310enyWT8c09B301kq0b1k",
      "name": "My key"
    }
  ]
}
```

- Per API i dettagli, vedere [GetApiKeys](#) in AWS CLI Command Reference.

## get-authorizer

Il seguente esempio di codice mostra come utilizzare `get-authorizer`.

AWS CLI

Per ottenere le impostazioni per ogni API autorizzazione del API Gateway

Comando:

```
aws apigateway get-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3
```

Output:

```
{
```

```
"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"  
}
```

- Per API i dettagli, vedere [GetAuthorizer](#) in AWS CLI Command Reference.

## get-authorizers

Il seguente esempio di codice mostra come utilizzare `get-authorizers`.

### AWS CLI

Per ottenere l'elenco degli autorizzatori per un REST API

Comando:

```
aws apigateway get-authorizers --rest-api-id 1234123412
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [GetAuthorizers](#) in AWS CLI Command Reference.

## get-base-path-mapping

Il seguente esempio di codice mostra come utilizzare `get-base-path-mapping`.

### AWS CLI

Per ottenere la mappatura del percorso di base per un nome di dominio personalizzato

Comando:

```
aws apigateway get-base-path-mapping --domain-name subdomain.domain.tld --base-path v1
```

Output:

```
{
  "basePath": "v1",
  "restApiId": "1234w4321e",
  "stage": "api"
}
```

- Per API i dettagli, vedere [GetBasePathMapping](#) in AWS CLI Command Reference.

## get-base-path-mappings

Il seguente esempio di codice mostra come utilizzare `get-base-path-mappings`.

### AWS CLI

Per ottenere le mappature del percorso di base per un nome di dominio personalizzato

Comando:

```
aws apigateway get-base-path-mappings --domain-name subdomain.domain.tld
```

Output:

```
{
  "items": [
    {
```

```
    "basePath": "(none)",
    "restApiId": "1234w4321e",
    "stage": "dev"
  },
  {
    "basePath": "v1",
    "restApiId": "1234w4321e",
    "stage": "api"
  }
]
```

- Per API i dettagli, vedere [GetBasePathMappings](#) in AWS CLI Command Reference.

## get-client-certificate

Il seguente esempio di codice mostra come utilizzare `get-client-certificate`.

### AWS CLI

Per ottenere un certificato client

Comando:

```
aws apigateway get-client-certificate --client-certificate-id a1b2c3
```

- Per API i dettagli, vedere [GetClientCertificate](#) in AWS CLI Command Reference.

## get-client-certificates

Il seguente esempio di codice mostra come utilizzare `get-client-certificates`.

### AWS CLI

Per ottenere un elenco di certificati client

Comando:

```
aws apigateway get-client-certificates
```

Output:

```
{
  "items": [
    {
      "pemEncodedCertificate": "-----BEGIN CERTIFICATE----- <certificate
content> -----END CERTIFICATE-----",
      "clientCertificateId": "a1b2c3",
      "expirationDate": 1483556561,
      "description": "My Client Certificate",
      "createdDate": 1452020561
    }
  ]
}
```

- Per API i dettagli, vedere [GetClientCertificates](#) in AWS CLI Command Reference.

## get-deployment

Il seguente esempio di codice mostra come utilizzare `get-deployment`.

### AWS CLI

Per ottenere informazioni su una distribuzione

Comando:

```
aws apigateway get-deployment --rest-api-id 1234123412 --deployment-id ztt4m2
```

Output:

```
{
  "description": "myDeployment",
  "id": "ztt4m2",
  "createdDate": 1455218022
}
```

- Per API i dettagli, vedere [GetDeployment](#) in AWS CLI Command Reference.

## get-deployments

Il seguente esempio di codice mostra come utilizzare `get-deployments`.

## AWS CLI

Per ottenere un elenco di distribuzioni per un REST API

Comando:

```
aws apigateway get-deployments --rest-api-id 1234123412
```

Output:

```
{
  "items": [
    {
      "createdDate": 1453797217,
      "id": "0a2b4c",
      "description": "Deployed my API for the first time"
    }
  ]
}
```

- Per API i dettagli, vedere [GetDeployments](#) in AWS CLI Command Reference.

## get-domain-name

Il seguente esempio di codice mostra come utilizzare `get-domain-name`.

## AWS CLI

Per ottenere informazioni su un nome di dominio personalizzato

Comando:

```
aws apigateway get-domain-name --domain-name api.domain.tld
```

Output:

```
{
  "domainName": "api.domain.tld",
  "distributionDomainName": "d1a2f3a4c5o6d.cloudfront.net",
  "certificateName": "uploadedCertificate",
}
```



```
"certificateUploadDate": 1462565487
}
```

- Per API i dettagli, consulta [GetDomainName AWS CLI Command Reference](#).

## get-domain-names

Il seguente esempio di codice mostra come utilizzare `get-domain-names`.

### AWS CLI

Per ottenere un elenco di nomi di dominio personalizzati

Comando:

```
aws apigateway get-domain-names
```

Output:

```
{
  "items": [
    {
      "distributionDomainName": "d9511k3109bkd.cloudfront.net",
      "certificateUploadDate": 1452812505,
      "certificateName": "my_custom_domain-certificate",
      "domainName": "subdomain.domain.tld"
    }
  ]
}
```

- Per API i dettagli, vedere [GetDomainNames](#) in AWS CLI Command Reference.

## get-export

Il seguente esempio di codice mostra come utilizzare `get-export`.

### AWS CLI

Per ottenere il modello JSON Swagger per uno stage

Comando:

```
aws apigateway get-export --rest-api-id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

Per ottenere il modello JSON Swagger + API Gateway Extensions per uno stage

Comando:

```
aws apigateway get-export --parameters extensions='integrations' --rest-api-id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

Per ottenere il modello JSON Swagger + Postman Extensions per un livello

Comando:

```
aws apigateway get-export --parameters extensions='postman' --rest-api-id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

- Per API i dettagli, consulta Command [GetExport](#)Reference AWS CLI .

## get-integration-response

Il seguente esempio di codice mostra come utilizzare `get-integration-response`.

AWS CLI

Per ottenere la configurazione della risposta di integrazione per un HTTP metodo definito REST API nella risorsa a

Comando:

```
aws apigateway get-integration-response --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET --status-code 200
```

Output:

```
{
  "statusCode": "200",
  "responseTemplates": {
    "application/json": null
  }
}
```

- Per API i dettagli, vedere [GetIntegrationResponse](#) in AWS CLI Command Reference.

## get-integration

Il seguente esempio di codice mostra come utilizzare `get-integration`.

### AWS CLI

Per ottenere la configurazione di integrazione per un HTTP metodo definito REST API nella risorsa a

Comando:

```
aws apigateway get-integration --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [GetIntegration](#) in AWS CLI Command Reference.

## get-method-response

Il seguente esempio di codice mostra come utilizzare `get-method-response`.

## AWS CLI

Per ottenere la configurazione della risorsa di risposta del metodo per un HTTP metodo definito nella risorsa REST API di a

Comando:

```
aws apigateway get-method-response --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET --status-code 200
```

Output:

```
{
  "responseModels": {
    "application/json": "Empty"
  },
  "statusCode": "200"
}
```

- Per API i dettagli, vedere [GetMethodResponse](#) in AWS CLI Command Reference.

## get-method

Il seguente esempio di codice mostra come utilizzare `get-method`.

## AWS CLI

Per ottenere la configurazione delle risorse del metodo per un HTTP metodo definito nella risorsa REST API di a

Comando:

```
aws apigateway get-method --rest-api-id 1234123412 --resource-id y9h6rt --http-method GET
```

Output:

```
{
  "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"
}

```

- Per API i dettagli, vedere [GetMethod](#) in AWS CLI Command Reference.

## get-model-template

Il seguente esempio di codice mostra come utilizzare `get-model-template`.

### AWS CLI

Per ottenere il modello di mappatura per un modello definito in un REST API

Comando:

```
aws apigateway get-model-template --rest-api-id 1234123412 --model-name Empty
```

Output:

```
{
```

```
"value": "#set($inputRoot = $input.path('$'))\n{ }"  
}
```

- Per API i dettagli, vedere [GetModelTemplate](#) in AWS CLI Command Reference.

## get-model

Il seguente esempio di codice mostra come utilizzare `get-model`.

### AWS CLI

Per ottenere la configurazione per un modello definito in un REST API

Comando:

```
aws apigateway get-model --rest-api-id 1234123412 --model-name Empty
```

Output:

```
{  
  "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}"  
}
```

- Per API i dettagli, vedere [GetModel](#) in AWS CLI Command Reference.

## get-models

Il seguente esempio di codice mostra come utilizzare `get-models`.

### AWS CLI

Per ottenere un elenco di modelli per un REST API

Comando:

```
aws apigateway get-models --rest-api-id 1234123412
```

**Output:**

```
{
  "items": [
    {
      "description": "This is a default error schema model",
      "schema": "{\n  \"$schema\" : \"http://json-schema.org/draft-04/schema#\n\",\n  \"title\" : \"Error Schema\",\n  \"type\" : \"object\",\n  \"properties\" : {\n    \"message\" : { \"type\" : \"string\" }\n  }\n}",
      "contentType": "application/json",
      "id": "7tpbze",
      "name": "Error"
    },
    {
      "description": "This is a default empty schema model",
      "schema": "{\n  \"$schema\" : \"http://json-schema.org/draft-04/schema#\n\",\n  \"title\" : \"Empty Schema\",\n  \"type\" : \"object\"\n}",
      "contentType": "application/json",
      "id": "etd5w5",
      "name": "Empty"
    }
  ]
}
```

- Per API i dettagli, vedere [GetModels](#) in AWS CLI Command Reference.

**get-resource**

Il seguente esempio di codice mostra come utilizzare `get-resource`.

**AWS CLI**

Per ottenere informazioni su una risorsa

Comando:

```
aws apigateway get-resource --rest-api-id 1234123412 --resource-id zwo0y3
```

Output:

```
{
```

```
"path": "/path",
"pathPart": "path",
"id": "zwo0y3",
"parentId": "uyokt6ij2g"
}
```

- Per API i dettagli, vedere [GetResource](#) in AWS CLI Command Reference.

## get-resources

Il seguente esempio di codice mostra come utilizzare `get-resources`.

### AWS CLI

Per ottenere un elenco di risorse per un REST API

Comando:

```
aws apigateway get-resources --rest-api-id 1234123412
```

Output:

```
{
  "items": [
    {
      "path": "/resource/subresource",
      "resourceMethods": {
        "POST": {}
      },
      "id": "024ace",
      "pathPart": "subresource",
      "parentId": "ai5b02"
    }
  ]
}
```

- Per API i dettagli, vedere [GetResources](#) in AWS CLI Command Reference.

## get-rest-api

Il seguente esempio di codice mostra come utilizzare `get-rest-api`.



## AWS CLI

Per ottenere informazioni su un API

Comando:

```
aws apigateway get-rest-api --rest-api-id 1234123412
```

Output:

```
{
  "name": "myAPI",
  "id": "o1y243m4f5",
  "createdDate": 1453416433
}
```

- Per API i dettagli, vedere [GetRestApi](#) in AWS CLI Command Reference.

## get-rest-apis

Il seguente esempio di codice mostra come utilizzare `get-rest-apis`.

## AWS CLI

Per ottenere un elenco di REST APIs

Comando:

```
aws apigateway get-rest-apis
```

Output:

```
{
  "items": [
    {
      "createdDate": 1438884790,
      "id": "12s44z21rb",
      "name": "My First API"
    }
  ]
}
```

```
}
```

- Per API i dettagli, vedere [GetRestApis](#) in AWS CLI Command Reference.

## get-sdk

Il seguente esempio di codice mostra come utilizzare `get-sdk`.

### AWS CLI

Per installare Android SDK per un REST API palco

Comando:

```
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
```

Output:

```
{
  "contentType": "application/octet-stream",
  "contentDisposition": "attachment; filename=\"android_2016-02-22_23-52Z.zip\""
}
```

Per procurarselo IOS SDK per un REST API palco

Comando:

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-
type objectivec --parameters classPrefix='myprefix' /path/to/iOS_sdk.zip
```

Output:

```
{
  "contentType": "application/octet-stream",
  "contentDisposition": "attachment; filename=\"objectivec_2016-02-22_23-52Z.zip
\""
}
```

Per ottenere il Javascript SDK per uno REST API stage

Comando:

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-type javascript /path/to/javascript_sdk.zip
```

Output:

```
{
  "contentType": "application/octet-stream",
  "contentDisposition": "attachment; filename=\"javascript_2016-02-22_23-52Z.zip\""}
}
```

- Per API i dettagli, vedere [GetSdk](#) in AWS CLI Command Reference.

## get-stage

Il seguente esempio di codice mostra come utilizzare `get-stage`.

AWS CLI

Per ottenere informazioni sullo stadio API di un

Comando:

```
aws apigateway get-stage --rest-api-id 1234123412 --stage-name dev
```

Output:

```
{
  "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
    }
  }
}
```

- Per API i dettagli, vedere [GetStage](#) in AWS CLI Command Reference.

## get-stages

Il seguente esempio di codice mostra come utilizzare `get-stages`.

### AWS CLI

Per ottenere l'elenco delle fasi di un REST API

Comando:

```
aws apigateway get-stages --rest-api-id 1234123412
```

**Output:**

```
{
  "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
        }
      }
    }
  ]
}
```

- Per API i dettagli, vedere [GetStages](#) in AWS CLI Command Reference.

**get-usage-plan-key**

Il seguente esempio di codice mostra come utilizzare `get-usage-plan-key`.

**AWS CLI**

Per ottenere i dettagli di una API chiave associata a un piano di utilizzo

Comando:

```
aws apigateway get-usage-plan-key --usage-plan-id a1b2c3 --key-id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- Per API i dettagli, vedere [GetUsagePlanKey](#) in AWS CLI Command Reference.

## get-usage-plan-keys

Il seguente esempio di codice mostra come utilizzare `get-usage-plan-keys`.

### AWS CLI

Per ottenere l'elenco delle API chiavi associate a un piano di utilizzo

Comando:

```
aws apigateway get-usage-plan-keys --usage-plan-id a1b2c3
```

- Per API i dettagli, vedere [GetUsagePlanKeys](#) in AWS CLI Command Reference.

## get-usage-plan

Il seguente esempio di codice mostra come utilizzare `get-usage-plan`.

### AWS CLI

Per ottenere i dettagli di un piano di utilizzo

Comando:

```
aws apigateway get-usage-plan --usage-plan-id a1b2c3
```

- Per API i dettagli, vedere [GetUsagePlan](#) in AWS CLI Command Reference.

## get-usage-plans

Il seguente esempio di codice mostra come utilizzare `get-usage-plans`.

### AWS CLI

Per ottenere i dettagli di tutti i piani di utilizzo

Comando:

```
aws apigateway get-usage-plans
```

- Per API i dettagli, vedere [GetUsagePlans](#) in AWS CLI Command Reference.

## get-usage

Il seguente esempio di codice mostra come utilizzare `get-usage`.

### AWS CLI

Per ottenere i dettagli di utilizzo di un piano di utilizzo

Comando:

```
aws apigateway get-usage --usage-plan-id a1b2c3 --start-date "2016-08-16" --end-date "2016-08-17"
```

- Per API i dettagli, vedere [GetUsage](#) in AWS CLI Command Reference.

## import-rest-api

Il seguente esempio di codice mostra come utilizzare `import-rest-api`.

### AWS CLI

Per importare un modello Swagger e creare un API

Comando:

```
aws apigateway import-rest-api --body 'file:///path/to/API_Swagger_template.json'
```

- Per API i dettagli, consulta AWS CLI Command [ImportRestApi](#) Reference.

## put-integration-response

Il seguente esempio di codice mostra come utilizzare `put-integration-response`.

### AWS CLI

Per creare una risposta di integrazione come risposta predefinita con un modello di mappatura definito

Comando:

```
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\"}"}'
```

Per creare una risposta di integrazione con un'espressione regolare di 400 e un valore di intestazione definito staticamente

Comando:

```
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": ""}'
```

- Per API i dettagli, vedere [PutIntegrationResponse](#) in AWS CLI Command Reference.

## put-integration

Il seguente esempio di codice mostra come utilizzare `put-integration`.

### AWS CLI

Per creare una richiesta di MOCK integrazione

Comando:

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-method GET --type MOCK --request-templates '{"application/json": "{\"statusCode\": 200}"}'
```

Per creare una richiesta di HTTP integrazione

Comando:

```
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'
```

Per creare una richiesta di AWS integrazione con un endpoint della funzione Lambda

Comando:



```
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'
```

- Per API i dettagli, consulta AWS CLI Command [PutIntegration](#) Reference.

## put-method-response

Il seguente esempio di codice mostra come utilizzare `put-method-response`.

### AWS CLI

Per creare una risposta al metodo con il codice di stato specificato con un'intestazione di risposta del metodo personalizzata

Comando:

```
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"
```

- Per API i dettagli, vedere [PutMethodResponse](#) in AWS CLI Command Reference.

## put-method

Il seguente esempio di codice mostra come utilizzare `put-method`.

### AWS CLI

Per creare un metodo per una risorsa in un'intestazione di richiesta API di metodo senza autorizzazione, senza API chiave e in un'intestazione di richiesta del metodo personalizzata

Comando:

```
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"
```

- Per API i dettagli, vedere [PutMethod](#) in AWS CLI Command Reference.

## put-rest-api

Il seguente esempio di codice mostra come utilizzare `put-rest-api`.

### AWS CLI

Sovrascrivere un file esistente API utilizzando un modello Swagger

Comando:

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode overwrite --body  
'fileb:///path/to/API_Swagger_template.json'
```

Per unire un modello Swagger a uno esistente API

Comando:

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode merge --body 'fileb:///  
path/to/API_Swagger_template.json'
```

- Per API i dettagli, consulta Command [PutRestApiReference](#) AWS CLI .

## test-invoke-authorizer

Il seguente esempio di codice mostra come utilizzare `test-invoke-authorizer`.

### AWS CLI

Per testare, richiama una richiesta a un Autorizzatore personalizzato che includa l'intestazione e il valore richiesti

Comando:

```
aws apigateway test-invoke-authorizer --rest-api-id 1234123412 --authorizer-  
id 5yid1t --headers Authorization='Value'
```

- Per API i dettagli, vedere [TestInvokeAuthorizer](#) in AWS CLI Command Reference.

## test-invoke-method

Il seguente esempio di codice mostra come utilizzare `test-invoke-method`.

## AWS CLI

Per testare, invoca la risorsa root in un API effettuando una GET richiesta.

Comando:

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id av15sg8fw8
--http-method GET --path-with-query-string '/'
```

Per testare, richiama una sottorisorsa in un API effettuando una GET richiesta con un valore del parametro di percorso specificato

Comando:

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id 3gapai --
http-method GET --path-with-query-string '/pets/1'
```

- Per API i dettagli, vedere [TestInvokeMethod](#) in AWS CLI Command Reference.

## update-account

Il seguente esempio di codice mostra come utilizzare `update-account`.

### AWS CLI

Per modificare il IAM ruolo ARN per la registrazione in Logs CloudWatch

Comando:

```
aws apigateway update-account --patch-operations op='replace',path='/
cloudwatchRoleArn',value='arn:aws:iam::123412341234:role/APIGatewayToCloudWatchLogs'
```

Output:

```
{
  "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/
APIGatewayToCloudWatchLogs",
  "throttleSettings": {
    "rateLimit": 1000.0,
    "burstLimit": 2000
  }
}
```

```
}  
}
```

- Per API i dettagli, vedere [UpdateAccount](#) in AWS CLI Command Reference.

## update-api-key

Il seguente esempio di codice mostra come utilizzare `update-api-key`.

### AWS CLI

Per cambiare il nome di una API chiave

Comando:

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --  
patch-operations op='replace',path='/name',value='newName'
```

Output:

```
{  
  "description": "currentDescription",  
  "enabled": true,  
  "stageKeys": [  
    "41t2j324r5/dev"  
  ],  
  "lastUpdatedDate": 1470086052,  
  "createdDate": 1445460347,  
  "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",  
  "name": "newName"  
}
```

Per disabilitare la API chiave

Comando:

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --  
patch-operations op='replace',path='/enabled',value='false'
```

Output:

```
{
  "description": "currentDescription",
  "enabled": false,
  "stageKeys": [
    "41t2j324r5/dev"
  ],
  "lastUpdatedDate": 1470086052,
  "createdDate": 1445460347,
  "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",
  "name": "newName"
}
```

- Per API i dettagli, vedere [UpdateApiKey](#) in AWS CLI Command Reference.

## update-authorizer

Il seguente esempio di codice mostra come utilizzare `update-authorizer`.

### AWS CLI

Per modificare il nome del Custom Authorizer

Comando:

```
aws apigateway update-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3 --
patch-operations op='replace',path='/name',value='testAuthorizer'
```

Output:

```
{
  "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"
}
```

Per modificare la funzione Lambda richiamata dall'Autoreizzatore personalizzato

Comando:

```
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'
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, consulta Command [UpdateAuthorizer](#)Reference AWS CLI .

## update-base-path-mapping

Il seguente esempio di codice mostra come utilizzare `update-base-path-mapping`.

AWS CLI

Per modificare il percorso di base per un nome di dominio personalizzato

Comando:

```
aws apigateway update-base-path-mapping --domain-name api.domain.tld --base-
path prod --patch-operations op='replace',path='/basePath',value='v1'
```

Output:

```
{
  "basePath": "v1",
  "restApiId": "1234123412",
  "stage": "api"
}
```

```
}
```

- Per API i dettagli, vedere [UpdateBasePathMapping](#) in AWS CLI Command Reference.

## update-client-certificate

Il seguente esempio di codice mostra come utilizzare `update-client-certificate`.

### AWS CLI

Per aggiornare la descrizione di un certificato client

Comando:

```
aws apigateway update-client-certificate --client-certificate-id a1b2c3 --patch-operations op='replace',path='/description',value='My new description'
```

- Per API i dettagli, vedere [UpdateClientCertificate](#) in AWS CLI Command Reference.

## update-deployment

Il seguente esempio di codice mostra come utilizzare `update-deployment`.

### AWS CLI

Per modificare la descrizione di una distribuzione

Comando:

```
aws apigateway update-deployment --rest-api-id 1234123412 --deployment-id ztt4m2 --patch-operations op='replace',path='/description',value='newDescription'
```

Output:

```
{
  "description": "newDescription",
  "id": "ztt4m2",
  "createdDate": 1455218022
}
```

- Per API i dettagli, vedere [UpdateDeployment](#) in AWS CLI Command Reference.

## update-domain-name

Il seguente esempio di codice mostra come utilizzare `update-domain-name`.

### AWS CLI

Per modificare il nome del certificato per un nome di dominio personalizzato

L'esempio seguente modifica il nome del certificato per un dominio personalizzato.

```
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'
```

Output:

```
{  
  "domainName": "api.domain.tld",  
  "distributionDomainName": "d123456789012.cloudfront.net",  
  "certificateArn": "arn:aws:acm:us-west-2:111122223333:certificate/  
CERTEXAMPLE123EXAMPLE",  
  "certificateUploadDate": 1462565487  
}
```

Per ulteriori informazioni, consulta [Configurare un nome di dominio personalizzato per un API in API Gateway nella Amazon API Gateway Developer Guide](#).

- Per API i dettagli, consulta [UpdateDomainName AWS CLI Command Reference](#).

## update-integration-response

Il seguente esempio di codice mostra come utilizzare `update-integration-response`.

### AWS CLI

Per modificare l'intestazione di una risposta di integrazione in modo da avere una mappatura statica di '\*'



Comando:

```
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='''''*''''
```

Output:

```
{  
  "statusCode": "200",  
  "responseParameters": {  
    "method.response.header.Access-Control-Allow-Origin": "''*''"  
  }  
}
```

Per rimuovere un'intestazione di risposta di integrazione

Comando:

```
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'
```

- Per API i dettagli, vedere [UpdateIntegrationResponse](#) in AWS CLI Command Reference.

## update-integration

Il seguente esempio di codice mostra come utilizzare `update-integration`.

AWS CLI

Per aggiungere il modello di mappatura 'Content-Type: application/json' configurato con Input Passthrough

Comando:

```
aws apigateway update-integration \  
  --rest-api-id a1b2c3d4e5 \  
  --resource-id a1b2c3 \  
  --http-method POST \  
  --patch-operations op='replace',path='/responseParameters/method.response.header.Content-Type,value='application/json'
```

```
--patch-operations "op='add',path='/requestTemplates/application~1json'"
```

Per aggiornare (sostituire) il modello di mappatura «Content-Type: application/json» configurato con un modello personalizzato

Comando:

```
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\"}'"
```

Per aggiornare (sostituire) un modello personalizzato associato a 'Content-Type: application/json' con Input Passthrough

Comando:

```
aws apigateway update-integration \  
  --rest-api-id a1b2c3d4e5 \  
  --resource-id a1b2c3 \  
  --http-method POST \  
  --patch-operations "op='replace',path='requestTemplates/application~1json'"
```

Per rimuovere il modello di mappatura 'Content-Type: application/json'

Comando:

```
aws apigateway update-integration \  
  --rest-api-id a1b2c3d4e5 \  
  --resource-id a1b2c3 \  
  --http-method POST \  
  --patch-operations "op='remove',path='/requestTemplates/application~1json'"
```

- Per i dettagli, vedere in Command Reference. API [UpdateIntegration](#) AWS CLI

## update-method-response

Il seguente esempio di codice mostra come utilizzare `update-method-response`.

## AWS CLI

Per creare una nuova intestazione di risposta al metodo per la risposta 200 in un metodo e definirla come non obbligatoria (impostazione predefinita)

Comando:

```
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"
```

Per eliminare un modello di risposta per la risposta 200 in un metodo

Comando:

```
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"
```

- Per API i dettagli, vedere [UpdateMethodResponse](#) in AWS CLI Command Reference.

## update-method

Il seguente esempio di codice mostra come utilizzare `update-method`.

### AWS CLI

Esempio 1: modificare un metodo per richiedere una API chiave

L'`update-method` seguente modifica il metodo per richiedere una API chiave.

```
aws apigateway update-method \  
  --rest-api-id 1234123412 \  
  --resource-id a1b2c3 \  
  --http-method GET \  
  --patch-operations op="replace",path="/apiKeyRequired",value="true"
```

Output:

```
{  
  "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": {}
        }
      }
    }
  }
}

```

Esempio 2: modificare un metodo per richiedere l'autorizzazione IAM

L'update-methodeseempio seguente modifica il metodo per richiedere IAM l'autorizzazione.

```

aws apigateway update-method \
  --rest-api-id 1234123412 \
  --resource-id a1b2c3 \
  --http-method GET \
  --patch-operations op="replace",path="/authorizationType",value="AWS_IAM"

```

Output:

```

{
  "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": {}
      }
    }
  }
}
}

```

### Esempio 3: modificare un metodo per richiedere l'autorizzazione Lambda

L'update-method esempio seguente modifica il metodo in base all'autorizzazione Lambda richiesta.

```

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"

```

Output:

```

{
  "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": {}
      }
    }
  }
}
}

```

Per ulteriori informazioni, consulta [Creare, configurare e testare i piani di utilizzo utilizzando il API Gateway CLI e Controllare REST API e gestire l'accesso a un API gateway REST API interno nella Amazon API Gateway Developer Guide](#).

- Per API i dettagli, consulta [UpdateMethod AWS CLI Command Reference](#).

## update-model

Il seguente esempio di codice mostra come utilizzare `update-model`.

### AWS CLI

Per modificare la descrizione di un modello in un API

Comando:

```
aws apigateway update-model --rest-api-id 1234123412 --model-name 'Empty' --patch-operations op=replace,path=/description,value='New Description'
```

Per modificare lo schema di un modello in un API

Comando:

```
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\" }"
```

- Per API i dettagli, vedere [UpdateModel](#) in AWS CLI Command Reference.

## update-resource

Il seguente esempio di codice mostra come utilizzare `update-resource`.

AWS CLI

Per spostare una risorsa e posizionarla sotto un'altra risorsa principale in un API

Comando:

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --patch-operations op=replace,path=/parentId,value='3c2b1a'
```

Output:

```
{
  "path": "/resource",
  "pathPart": "resource",
  "id": "1a2b3c",
  "parentId": "3c2b1a"
}
```

Per rinominare una risorsa (pathPart) in un API

Comando:

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --patch-operations op=replace,path=/pathPart,value=newresourceName
```

Output:

```
{
  "path": "/newresourcenam",
  "pathPart": "newresourcenam",
  "id": "1a2b3c",
  "parentId": "3c2b1a"
}
```

- Per API i dettagli, vedere [UpdateResource](#) in AWS CLI Command Reference.

## update-rest-api

Il seguente esempio di codice mostra come utilizzare `update-rest-api`.

### AWS CLI

Per cambiare il nome di un API

Comando:

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
op=replace,path=/name,value='New Name'
```

Per modificare la descrizione di un API

Comando:

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
op=replace,path=/description,value='New Description'
```

- Per API i dettagli, vedere [UpdateRestApi](#) in AWS CLI Command Reference.

## update-stage

Il seguente esempio di codice mostra come utilizzare `update-stage`.

### AWS CLI

Esempio 1: Per sovrascrivere le impostazioni dello stage per una risorsa e un metodo

L'`update-stage` esempio seguente sovrascrive le impostazioni dello stage e disattiva la registrazione completa delle richieste/risposte per una risorsa e un metodo specifici.



```
aws apigateway update-stage \  
  --rest-api-id 1234123412 \  
  --stage-name 'dev' \  
  --patch-operations op=replace,path=~1resourceName/GET/Logging/  
dataTrace,value=false
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Configurazione di una fase per un REST API](#) nella Amazon API Gateway Developer Guide.

Esempio 2: Per aggiornare le impostazioni dello stage per tutte le risorse e i metodi di una API fase

L'update-stageesempio seguente attiva la registrazione completa delle richieste/risposte per tutte le risorse e i metodi di una fase. API

```
aws apigateway update-stage \  
  --rest-api-id 1234123412 \  
  --patch-operations op=replace,path=~1resourceName/GET/Logging/  
dataTrace,value=true
```

```
--stage-name 'dev' \  
--patch-operations 'op=replace,path=/**/logging/dataTrace,value=true'
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Configurazione di una fase per un REST API](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [UpdateStage AWS CLI Command Reference](#).

## update-usage-plan

Il seguente esempio di codice mostra come utilizzare `update-usage-plan`.

### AWS CLI

Per modificare il periodo definito in un piano di utilizzo

Comando:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/quota/period",value="MONTH"
```

Per modificare il limite di quota definito in un piano di utilizzo

Comando:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/quota/limit",value="500"
```

Per modificare il limite di velocità di accelerazione definito in un piano di utilizzo

Comando:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/throttle/rateLimit",value="10"
```

Per modificare il limite di accelerazione definito in un piano di utilizzo

Comando:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations  
op="replace",path="/throttle/burstLimit",value="20"
```

- Per API i dettagli, consulta Command [UpdateUsagePlanReference AWS CLI](#) .

## update-usage

Il seguente esempio di codice mostra come utilizzare `update-usage`.

AWS CLI

Per modificare temporaneamente la quota su una API chiave per il periodo corrente definito nel piano di utilizzo

Comando:

```
aws apigateway update-usage --usage-plan-id a1b2c3 --key-  
id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu --patch-operations op="replace",path="/  
remaining",value="50"
```

- Per API i dettagli, vedere [UpdateUsage](#) in AWS CLI Command Reference.

## APIGateway HTTP ed WebSocket API esempi di utilizzo AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with API Gateway HTTP and WebSocket API.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-api-mapping**

Il seguente esempio di codice mostra come utilizzare `create-api-mapping`.

AWS CLI

Per creare una API mappatura per un API

L'`create-api-mapping` esempio seguente mappa lo test stadio di un API al `/myApi` percorso del nome di dominio `regional.example.com` personalizzato.

```
aws apigatewayv2 create-api-mapping \  
  --domain-name regional.example.com \  
  --api-mapping-key myApi \  
  --api-id a1b2c3d4 \  
  --stage test
```

Output:

```
{
```

```
"ApiId": "a1b2c3d4",
"ApiMappingId": "0qzs2sy7bh",
"ApiMappingKey": "myApi"
"Stage": "test"
}
```

Per ulteriori informazioni, consulta [Configurazione di un nome di dominio personalizzato regionale in API Gateway nella Amazon API Gateway Developer Guide](#).

- Per API i dettagli, consulta [CreateApiMapping AWS CLI Command Reference](#).

## create-api

Il seguente esempio di codice mostra come utilizzare `create-api`.

### AWS CLI

Per creare un HTTP API

L'`create-api` esempio seguente crea un HTTP API utilizzando la creazione rapida. Puoi utilizzare la creazione rapida per creare un percorso API con AWS Lambda o HTTP integrazione, un percorso generico predefinito e uno stadio predefinito configurato per distribuire automaticamente le modifiche. Il comando seguente utilizza la creazione rapida per creare un file HTTP API che si integra con una funzione Lambda.

```
aws apigatewayv2 create-api \
  --name my-http-api \
  --protocol-type HTTP \
  --target arn:aws:lambda:us-west-2:123456789012:function:my-lambda-function
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Developing an HTTP API in API Gateway](#) nella Amazon API Gateway Developer Guide.

Per creare un WebSocket API

L'create-api esempio seguente crea un file WebSocket API con il nome specificato.

```
aws apigatewayv2 create-api \  
  --name "myWebSocketApi" \  
  --protocol-type WEBSOCKET \  
  --route-selection-expression '$request.body.action'
```

Output:

```
{  
  "ApiKeySelectionExpression": "$request.header.x-api-key",  
  "Name": "myWebSocketApi",  
  "CreateDate": "2018-11-15T06:23:51Z",  
  "ProtocolType": "WEBSOCKET",  
  "RouteSelectionExpression": "'$request.body.action'",  
  "ApiId": "aabbccdde"  
}
```

Per ulteriori informazioni, consulta [Create a WebSocket API in API Gateway](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [CreateApi AWS CLI Command Reference](#).

## create-authorizer

Il seguente esempio di codice mostra come utilizzare create-authorizer.

### AWS CLI

Per creare un JWT autorizzatore per un HTTP API

L'create-authorizer esempio seguente crea un JWT autorizzatore che utilizza Amazon Cognito come provider di identità.

```
aws apigatewayv2 create-authorizer \  
  --name "myAuthorizer" \  
  --protocol-type HTTP \  
  --route-selection-expression '$request.body.action'
```

```
--name my-jwt-authorizer \  
--api-id a1b2c3d4 \  
--authorizer-type JWT \  
--identity-source '$request.header.Authorization' \  
--jwt-configuration Audience=123456abc,Issuer=https://cognito-idp.us-west-2.amazonaws.com/us-west-2_abc123
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Controlling access to HTTP APIs with JWT authorizers](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [CreateAuthorizer AWS CLI Command Reference](#).

## create-deployment

Il seguente esempio di codice mostra come utilizzare `create-deployment`.

### AWS CLI

Per creare una distribuzione per un API

L'esempio seguente crea una distribuzione per un API e associa tale distribuzione alla dev fase di API

```
aws apigatewayv2 create-deployment \  
--api-id a1b2c3d4 \  
--stage-name dev \  
--deployment-name my-deployment
```

```
--stage-name dev
```

Output:

```
{
  "AutoDeployed": false,
  "CreateDate": "2020-04-06T23:38:08Z",
  "DeploymentId": "531z91",
  "DeploymentStatus": "DEPLOYED"
}
```

Per ulteriori informazioni, consulta la sezione relativa alla [API distribuzione](#) nell'Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [CreateDeployment AWS CLI Command Reference](#).

## create-domain-name

Il seguente esempio di codice mostra come utilizzare `create-domain-name`.

### AWS CLI

Per creare un nome di dominio personalizzato

L'esempio seguente crea un nome di dominio personalizzato regionale per un'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
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Configurazione di un nome di dominio personalizzato regionale in API Gateway nella Amazon API Gateway Developer Guide](#).

- Per API i dettagli, consulta [CreateDomainName AWS CLI Command Reference](#).

## create-integration

Il seguente esempio di codice mostra come utilizzare `create-integration`.

### AWS CLI

Per creare un' WebSocket API integrazione

L'`create-integration` esempio seguente crea un'integrazione fittizia per un WebSocket API.

```

aws apigatewayv2 create-integration \
  --api-id aabbccdee \
  --passthrough-behavior WHEN_NO_MATCH \
  --timeout-in-millis 29000 \
  --connection-type INTERNET \
  --integration-type MOCK

```

Output:

```

{
  "ConnectionType": "INTERNET",
  "IntegrationId": "0abcdef",
  "IntegrationResponseSelectionExpression": "${integration.response.statuscode}",
  "IntegrationType": "MOCK",
  "PassthroughBehavior": "WHEN_NO_MATCH",
  "PayloadFormatVersion": "1.0",
  "TimeoutInMillis": 29000
}

```

Per ulteriori informazioni, consulta [Configurare una richiesta di WebSocket API integrazione in API Gateway nella](#) Amazon API Gateway Developer Guide.

Per creare un'HTTPAPIintegrazione

L'create-integrationesempio seguente crea un'integrazione AWS Lambda per un. HTTP API

```
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
```

Output:

```
{  
  "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  
}
```

Per ulteriori informazioni, consulta la sezione [Configurazione delle integrazioni HTTP APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateIntegrationReference](#).

## create-route

Il seguente esempio di codice mostra come utilizzare create-route.

AWS CLI

Per creare una route \$default per un o WebSocket HTTP API

L'create-routeesempio seguente crea una \$default rotta per un WebSocket o HTTPAPI.

```
aws apigatewayv2 create-route \  
  --api-id a1b2c3d4 \  
  --integration-type AWS_PROXY \  
  --integration-uri arn:aws:lambda:us-west-2:123456789012:function:my-function \  
  --payload-format-version 2.0
```

```
--api-id aabbccdde \  
--route-key '$default'
```

Output:

```
{  
  "ApiKeyRequired": false,  
  "AuthorizationType": "NONE",  
  "RouteKey": "$default",  
  "RouteId": "1122334"  
}
```

Per ulteriori informazioni, consulta [Working with routes for WebSocket APIs](#) nella Amazon API Gateway Developer Guide

Per creare un percorso per un HTTP API

L'create-routeesempio seguente crea una rotta denominata `signup` che accetta POST richieste.

```
aws apigatewayv2 create-route \  
  --api-id aabbccdde \  
  --route-key 'POST /signup'
```

Output:

```
{  
  "ApiKeyRequired": false,  
  "AuthorizationType": "NONE",  
  "RouteKey": "POST /signup",  
  "RouteId": "1122334"  
}
```

Per ulteriori informazioni, consulta [Working with routes for HTTP APIs](#) nella Amazon API Gateway Developer Guide

- Per API i dettagli, consulta [CreateRoute AWS CLI Command Reference](#).

## create-stage

Il seguente esempio di codice mostra come utilizzare `create-stage`.

## AWS CLI

Per creare uno stage

L'`create-stage` seguente crea una fase denominata `dev` per un API.

```
aws apigatewayv2 create-stage \  
  --api-id a1b2c3d4 \  
  --stage-name dev
```

Output:

```
{  
  "CreateDate": "2020-04-06T23:23:46Z",  
  "DefaultRouteSettings": {  
    "DetailedMetricsEnabled": false  
  },  
  "LastUpdatedDate": "2020-04-06T23:23:46Z",  
  "RouteSettings": {},  
  "StageName": "dev",  
  "StageVariables": {},  
  "Tags": {}  
}
```

Per ulteriori informazioni, consulta [Working with stages for HTTP APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [CreateStage AWS CLI Command Reference](#).

## `create-vpc-link`

Il seguente esempio di codice mostra come utilizzare `create-vpc-link`.

## AWS CLI

Per creare un VPC collegamento per un HTTP API

L'`create-vpc-link` seguente crea un VPC collegamento per HTTP APIs.

```
aws apigatewayv2 create-vpc-link \  
  --name MyVpcLink \  
  --stage-name dev
```

```
--subnet-ids subnet-aaaa subnet-bbbb \  
--security-group-ids sg1234 sg5678
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Working with VPC links for HTTP APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [CreateVpcLink AWS CLI Command Reference](#).

## delete-access-log-settings

Il seguente esempio di codice mostra come utilizzare `delete-access-log-settings`.

### AWS CLI

Per disabilitare la registrazione degli accessi per un API

L'esempio seguente elimina le impostazioni del registro di accesso per la `default` fase di un API. Per disabilitare la registrazione degli accessi per una fase, eliminate le relative impostazioni del registro di accesso.

```
aws apigatewayv2 delete-access-log-settings \  
--api-id a1b2c3d4 \  

```

```
--stage-name '$default'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configuring logging for an HTTP API](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteAccessLogSettings](#) Reference.

## delete-api-mapping

Il seguente esempio di codice mostra come utilizzare `delete-api-mapping`.

AWS CLI

Per eliminare una API mappatura

L'`delete-api-mapping` seguente elimina una API mappatura per il nome di dominio `api.example.com` personalizzato.

```
aws apigatewayv2 delete-api-mapping \  
  --api-mapping-id a1b2c3 \  
  --domain-name api.example.com
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configurazione di un nome di dominio personalizzato regionale in API Gateway nella](#) Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [DeleteApiMapping AWS CLI](#) Command Reference.

## delete-api

Il seguente esempio di codice mostra come utilizzare `delete-api`.

AWS CLI

Per eliminare un API

L'`delete-api` seguente elimina un API.

```
aws apigatewayv2 delete-api \  
  --api-id api-id
```

```
--api-id a1b2c3d4
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Working with HTTP APIs](#) and [Working with WebSocket APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [DeleteApi AWS CLI](#) Command Reference.

## **delete-authorizer**

Il seguente esempio di codice mostra come utilizzare `delete-authorizer`.

AWS CLI

Per eliminare un autorizzatore

L'`delete-authorizer` esempio seguente elimina un autorizzatore.

```
aws apigatewayv2 delete-authorizer \  
  --api-id a1b2c3d4 \  
  --authorizer-id a1b2c3
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Controlling access to HTTP APIs with JWT authorizers](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [DeleteAuthorizer AWS CLI](#) Command Reference.

## **delete-cors-configuration**

Il seguente esempio di codice mostra come utilizzare `delete-cors-configuration`.

AWS CLI

Per eliminare la CORS configurazione di un HTTP API

L'`delete-cors-configuration` esempio seguente disabilita CORS for an HTTP API eliminandone la CORS configurazione.

```
aws apigatewayv2 delete-cors-configuration \  
  --api-id a1b2c3d4 \  
  --stage-name prod
```

```
--api-id a1b2c3d4
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configuring CORS for an HTTP API](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [DeleteCorsConfiguration AWS CLI](#) Command Reference.

## delete-deployment

Il seguente esempio di codice mostra come utilizzare `delete-deployment`.

AWS CLI

Per eliminare una distribuzione

L'`delete-deployment` esempio seguente elimina una distribuzione di un API.

```
aws apigatewayv2 delete-deployment \  
  --api-id a1b2c3d4 \  
  --deployment-id a1b2c3
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione relativa alla [API distribuzione](#) nell'Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [DeleteDeployment AWS CLI](#) Command Reference.

## delete-domain-name

Il seguente esempio di codice mostra come utilizzare `delete-domain-name`.

AWS CLI

Per eliminare un nome di dominio personalizzato

L'`delete-domain-name` esempio seguente elimina un nome di dominio personalizzato.

```
aws apigatewayv2 delete-domain-name \  
  --domain-name api.example.com
```



Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configurazione di un nome di dominio personalizzato regionale in API Gateway nella](#) Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [DeleteDomainName AWS CLI](#) Command Reference.

## **delete-integration**

Il seguente esempio di codice mostra come utilizzare `delete-integration`.

### AWS CLI

Per eliminare un'integrazione

L'`delete-integration` esempio seguente elimina un'API integrazione.

```
aws apigatewayv2 delete-integration \  
  --api-id a1b2c3d4 \  
  --integration-id a1b2c3
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configurazione delle integrazioni HTTP APIs](#) e [Configurazione delle WebSocket API integrazioni](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta Command [DeleteIntegration](#) Reference AWS CLI .

## **delete-route-settings**

Il seguente esempio di codice mostra come utilizzare `delete-route-settings`.

### AWS CLI

Per eliminare le impostazioni del percorso

L'`delete-route-settings` esempio seguente elimina le impostazioni del percorso per il percorso specificato.

```
aws apigatewayv2 delete-route-settings \  
  --api-id a1b2c3d4 \  
  --stage-name dev \  
  --route-id 12345678
```

```
--route-key 'GET /pets'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Working with routes for HTTP APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [DeleteRouteSettings AWS CLI Command Reference](#).

## delete-route

Il seguente esempio di codice mostra come utilizzare `delete-route`.

AWS CLI

Per eliminare un percorso

L'`delete-route` esempio seguente elimina un API percorso.

```
aws apigatewayv2 delete-route \  
  --api-id a1b2c3d4 \  
  --route-id a1b2c3
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Working with routes for HTTP APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [DeleteRoute AWS CLI Command Reference](#).

## delete-stage

Il seguente esempio di codice mostra come utilizzare `delete-stage`.

AWS CLI

Per eliminare una fase

L'`delete-stage` esempio seguente elimina lo test stadio di un API.

```
aws apigatewayv2 delete-stage \  
  --api-id a1b2c3d4 \  
  --stage-name test
```

```
--api-id a1b2c3d4 \  
--stage-name test
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Working with stages for HTTP APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [DeleteStage AWS CLI](#) Command Reference.

## delete-vpc-link

Il seguente esempio di codice mostra come utilizzare `delete-vpc-link`.

### AWS CLI

Per eliminare un VPC collegamento per un HTTP API

L'`delete-vpc-link` esempio seguente elimina un VPC collegamento.

```
aws apigatewayv2 delete-vpc-link \  
--vpc-link-id abcd123
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Working with VPC links for HTTP APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [DeleteVpcLink AWS CLI](#) Command Reference.

## export-api

Il seguente esempio di codice mostra come utilizzare `export-api`.

### AWS CLI

Per esportare una API definizione aperta di un HTTP API

L'`export-api` esempio seguente esporta una definizione Open API 3.0 di una API fase denominata `prod` in un YAML file denominato `stage-definition.yaml`. Per impostazione predefinita, il file di definizione esportato include le estensioni API Gateway.

```
aws apigatewayv2 export-api \  
  --api-id a1b2c3d4 \  
  --output-type YAML \  
  --specification OAS30 \  
  --stage-name prod \  
  stage-definition.yaml
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Esportazione di un file HTTP API da API Gateway](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [ExportApi AWS CLI](#) Command Reference.

## get-api-mapping

Il seguente esempio di codice mostra come utilizzare `get-api-mapping`.

### AWS CLI

Per ottenere informazioni su una API mappatura per un nome di dominio personalizzato

L'`get-api-mapping` esempio seguente visualizza informazioni su una API mappatura per il `api.example.com` nome di dominio personalizzato.

```
aws apigatewayv2 get-api-mapping \  
  --api-mapping-id a1b2c3 \  
  --domain-name api.example.com
```

Output:

```
{  
  "ApiId": "a1b2c3d4",  
  "ApiMappingId": "a1b2c3d5",  
  "ApiMappingKey": "myTestApi"  
  "Stage": "test"  
}
```

Per ulteriori informazioni, consulta [Configurazione di un nome di dominio personalizzato regionale in API Gateway](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [GetApiMapping AWS CLI](#) Command Reference.

## get-api-mappings

Il seguente esempio di codice mostra come utilizzare `get-api-mappings`.

### AWS CLI

Per ottenere API mappature per un nome di dominio personalizzato

L'esempio seguente visualizza un elenco di tutte le API mappature per il `api.example.com` nome di dominio personalizzato.

```
aws apigatewayv2 get-api-mappings \  
  --domain-name api.example.com
```

Output:

```
{  
  "Items": [  
    {  
      "ApiId": "a1b2c3d4",  
      "ApiMappingId": "a1b2c3d5",  
      "ApiMappingKey": "myTestApi"  
      "Stage": "test"  
    },  
    {  
      "ApiId": "a5b6c7d8",  
      "ApiMappingId": "a1b2c3d6",  
      "ApiMappingKey": "myDevApi"  
      "Stage": "dev"  
    },  
  ]  
}
```

Per ulteriori informazioni, consulta [Configurazione di un nome di dominio personalizzato regionale in API Gateway nella Amazon API Gateway Developer Guide](#).

- Per API i dettagli, consulta [GetApiMappings AWS CLI Command Reference](#).

## get-api

Il seguente esempio di codice mostra come utilizzare `get-api`.

## AWS CLI

Per recuperare informazioni su un API

L'get -apiesempio seguente visualizza informazioni su unAPI.

```
aws apigatewayv2 get-api \  
  --api-id a1b2c3d4
```

Output:

```
{  
  "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"  
  }  
}
```

- Per API i dettagli, vedere [GetApi](#) in AWS CLI Command Reference.

## get-apis

Il seguente esempio di codice mostra come utilizzareget-apis.

## AWS CLI

Per recuperare un elenco di APIs

L'get -apisesempio seguente elenca tutti i dati APIs relativi all'utente corrente.

```
aws apigatewayv2 get-apis
```

Output:

```
{  
  "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": {}
    }
  ]
}

```

Per ulteriori informazioni, consulta [Working with HTTP APIs](#) and [Working with WebSocket APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [GetApis AWS CLI Command Reference](#).

## get-authorizer

Il seguente esempio di codice mostra come utilizzare `get-authorizer`.

### AWS CLI

Per recuperare informazioni su un autorizzatore

L'`get-authorizer` esempio seguente visualizza informazioni su un autorizzatore.

```

aws apigatewayv2 get-authorizer \
  --api-id a1b2c3d4 \
  --authorizer-id a1b2c3

```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Controlling access to HTTP APIs with JWT authorizers](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [GetAuthorizer AWS CLI Command Reference](#).

## get-authorizers

Il seguente esempio di codice mostra come utilizzare `get-authorizers`.

### AWS CLI

Per recuperare un elenco di autorizzatori per un API

L'`get-authorizers` esempio seguente visualizza un elenco di tutti gli autorizzatori per un API

```
aws apigatewayv2 get-authorizers \
  --api-id a1b2c3d4
```

Output:

```
{
  "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"
}
]
}

```

Per ulteriori informazioni, consulta [Controlling access to HTTP APIs with JWT authorizers](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [GetAuthorizers AWS CLI](#) Command Reference.

## get-deployment

Il seguente esempio di codice mostra come utilizzare `get-deployment`.

### AWS CLI

Per recuperare informazioni su una distribuzione

L'`get-deployment` esempio seguente visualizza informazioni su una distribuzione.

```
aws apigatewayv2 get-deployment \  
  --api-id a1b2c3d4 \  
  --deployment-id abcdef
```

Output:

```
{  
  "AutoDeployed": true,  
  "CreateDate": "2020-04-07T23:58:40Z",  
  "DeploymentId": "abcdef",  
  "DeploymentStatus": "DEPLOYED",  
  "Description": "Automatic deployment triggered by changes to the Api  
  configuration"  
}
```

Per ulteriori informazioni, consulta la sezione relativa alla [API distribuzione](#) nell'Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [GetDeployment AWS CLI Command Reference](#).

## get-deployments

Il seguente esempio di codice mostra come utilizzare `get-deployments`.

AWS CLI

Per recuperare un elenco di distribuzioni

L'`get-deployment` esempio seguente mostra un elenco di tutte le distribuzioni di un API.

```
aws apigatewayv2 get-deployments \  
  --api-id a1b2c3d4
```

Output:

```
{  
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta la sezione relativa alla [API distribuzione](#) nell'Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [GetDeployments AWS CLI Command Reference](#).

## get-domain-name

Il seguente esempio di codice mostra come utilizzare `get-domain-name`.

### AWS CLI

Per recuperare informazioni su un nome di dominio personalizzato

L'`get-domain-name` esempio seguente visualizza informazioni su un nome di dominio personalizzato.

```
aws apigatewayv2 get-domain-name \
  --domain-name api.example.com
```

Output:

```

{
  "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": {}
}

```

Per ulteriori informazioni, consulta [Configurazione di un nome di dominio personalizzato regionale in API Gateway nella Amazon API Gateway Developer Guide](#).

- Per API i dettagli, consulta [GetDomainName AWS CLI Command Reference](#).

## get-domain-names

Il seguente esempio di codice mostra come utilizzare `get-domain-names`.

### AWS CLI

Per recuperare un elenco di nomi di dominio personalizzati

L'`get-domain-names` esempio seguente visualizza un elenco di tutti i nomi di dominio personalizzati per l'utente corrente.

```
aws apigatewayv2 get-domain-names
```

Output:

```

{
  "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"
        }
    ]
}
]
}
}

```

Per ulteriori informazioni, consulta [Configurazione di un nome di dominio personalizzato regionale in API Gateway nella Amazon API Gateway Developer Guide](#).

- Per API i dettagli, consulta [GetDomainNames AWS CLI Command Reference](#).

## get-integration

Il seguente esempio di codice mostra come utilizzare `get-integration`.

### AWS CLI

Per recuperare informazioni su un'integrazione

L'`get-integration` esempio seguente mostra informazioni su un'integrazione.

```
aws apigatewayv2 get-integration \
  --api-id a1b2c3d4 \
```

```
--integration-id a1b2c3
```

Output:

```
{
  "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
}
```

Per ulteriori informazioni, consulta [Configurazione delle integrazioni HTTP APIs](#) e [Configurazione delle WebSocket API integrazioni](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta Command [GetIntegration](#)Reference AWS CLI .

## get-integrations

Il seguente esempio di codice mostra come utilizzare `get-integrations`.

AWS CLI

Per recuperare un elenco di integrazioni

L'`get-integrations` esempio seguente mostra un elenco di tutte le integrazioni API di an.

```
aws apigatewayv2 get-integrations \
  --api-id a1b2c3d4
```

Output:

```
{
  "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
  }
]
}

```

Per ulteriori informazioni, consulta [Configurazione delle integrazioni HTTP APIs](#) e [Configurazione delle WebSocket API integrazioni](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta Command [GetIntegrations](#) Reference AWS CLI .

## get-route

Il seguente esempio di codice mostra come utilizzare `get-route`.

### AWS CLI

Per recuperare informazioni su un percorso

L'`get-route` esempio seguente visualizza informazioni su un percorso.

```

aws apigatewayv2 get-route \
  --api-id a1b2c3d4 \
  --route-id 72jz1wk

```

Output:

```

{
  "ApiKeyRequired": false,
  "AuthorizationType": "NONE",

```

```
"RouteId": "72jz1wk",
"RouteKey": "ANY /pets",
"Target": "integrations/a1b2c3"
}
```

Per ulteriori informazioni, consulta [Working with routes for HTTP APIs](#) nella Amazon API Gateway Developer Guide

- Per API i dettagli, consulta [GetRoute AWS CLI Command Reference](#).

## get-routes

Il seguente esempio di codice mostra come utilizzare `get-routes`.

### AWS CLI

Per recuperare un elenco di percorsi

L'`get-routes` esempio seguente mostra un elenco di tutti i percorsi API di an.

```
aws apigatewayv2 get-routes \
  --api-id a1b2c3d4
```

Output:

```
{
  "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"
    }
  ]
}
```



```
}
```

Per ulteriori informazioni, consulta [Working with routes for HTTP APIs](#) nella Amazon API Gateway Developer Guide

- Per API i dettagli, consulta [GetRoutes AWS CLI Command Reference](#).

## get-stage

Il seguente esempio di codice mostra come utilizzare `get-stage`.

### AWS CLI

Per recuperare informazioni su una fase

L'`get-stage` esempio seguente visualizza informazioni sullo `prod` stadio di un'API.

```
aws apigatewayv2 get-stage \  
  --api-id a1b2c3d4 \  
  --stage-name prod
```

Output:

```
{  
  "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": {}  
}
```

Per ulteriori informazioni, consulta [Working with stages for HTTP APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [GetStage AWS CLI Command Reference](#).

## get-stages

Il seguente esempio di codice mostra come utilizzare `get-stages`.

### AWS CLI

Per recuperare un elenco di fasi

L'`get-stages` seguente elenca tutte le fasi API di an.

```
aws apigatewayv2 get-stages \  
  --api-id a1b2c3d4
```

Output:

```
{  
  "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": {}
}
]
}

```

Per ulteriori informazioni, consulta [Working with stages for HTTP APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [GetStages AWS CLI Command Reference](#).

## get-tags

Il seguente esempio di codice mostra come utilizzare `get-tags`.

### AWS CLI

Per recuperare un elenco di tag per una risorsa

L'`get-tags` esempio seguente elenca tutti i tag API di an.

```
aws apigatewayv2 get-tags \
  --resource-arn arn:aws:apigateway:us-west-2:/apis/a1b2c3d4
```

Output:

```
{
```

```
"Tags": {
  "owner": "dev-team",
  "environment": "prod"
}
```

Per ulteriori informazioni, consulta [Tagging your API Gateway resources](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [GetTags AWS CLI Command Reference](#).

## get-vpc-link

Il seguente esempio di codice mostra come utilizzare `get-vpc-link`.

### AWS CLI

Per recuperare informazioni su un collegamento VPC

L'`get-vpc-link` esempio seguente visualizza informazioni su un VPC collegamento.

```
aws apigatewayv2 get-vpc-link \
  --vpc-link-id abcd123
```

Output:

```
{
  "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"
```

```
}
```

Per ulteriori informazioni, consulta [Working with VPC links for HTTP APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [GetVpcLink AWS CLI Command Reference](#).

## get-vpc-links

Il seguente esempio di codice mostra come utilizzare `get-vpc-links`.

### AWS CLI

Per recuperare un elenco di collegamenti VPC

L'`get-vpc-links`esempio seguente visualizza un elenco di tutti i VPC link per l'utente corrente.

```
aws apigatewayv2 get-vpc-links
```

Output:

```
{
  "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"
  }
]
```

Per ulteriori informazioni, consulta [Working with VPC links for HTTP APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [GetVpcLinks AWS CLI Command Reference](#).

## import-api

Il seguente esempio di codice mostra come utilizzare `import-api`.

### AWS CLI

Per importare un HTTP API

L'esempio seguente crea un file HTTP API di definizione Open API 3.0 denominato `api-definition.yaml`.

```
aws apigatewayv2 import-api \
  --body file://api-definition.yaml
```

Contenuto di `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
```

### Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Working with Open definition API HTTP APIs nella Amazon API Gateway Developer Guide](#).

- Per API i dettagli, consulta [ImportApi AWS CLI Command Reference](#).

## reimport-api

Il seguente esempio di codice mostra come utilizzare `reimport-api`.

### AWS CLI

Per reimportare un HTTP API

L'`reimport-api` esempio seguente aggiorna un esistente HTTP API per utilizzare la definizione Open API 3.0 specificata in `api-definition.yaml`.

```
aws apigatewayv2 reimport-api \
```

```
--body file://api-definition.yaml \  
--api-id a1b2c3d4
```

Contenuto di `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
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Working with Open definition API HTTP APIs nella Amazon API Gateway Developer Guide](#).

- Per API i dettagli, consulta [ReimportApi AWS CLI Command Reference](#).

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.



## AWS CLI

Per etichettare una risorsa

L'`tag-resource` esempio seguente aggiunge un tag con il nome della chiave `Department` e `Accounting` il valore specificato API.

```
aws apigatewayv2 tag-resource \  
  --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4 \  
  --tags Department=Accounting
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging your API Gateway resources](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [TagResource AWS CLI Command Reference](#).

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

## AWS CLI

Per rimuovere i tag da una risorsa

L'`untag-resource` esempio seguente rimuove i tag con i nomi delle chiavi `Project` e `Owner` da quelli specificati API.

```
aws apigatewayv2 untag-resource \  
  --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4 \  
  --tag-keys Project Owner
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging your API Gateway resources](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [UntagResource AWS CLI Command Reference](#).

## update-api-mapping

Il seguente esempio di codice mostra come utilizzare `update-api-mapping`.

## AWS CLI

Per aggiornare una API mappatura

L'update-api-mappingesempio seguente modifica una API mappatura per un nome di dominio personalizzato. Di conseguenza, la base che URL utilizza il nome di dominio personalizzato per lo stadio API e specificato diventahttps://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
```

Output:

```
{  
  "ApiId": "a1b2c3d4",  
  "ApiMappingId": "0qzs2sy7bh",  
  "ApiMappingKey": "dev"  
  "Stage": "dev"  
}
```

Per ulteriori informazioni, consulta [Configurazione di un nome di dominio personalizzato regionale in API Gateway nella](#) Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [UpdateApiMapping AWS CLI](#) Command Reference.

## update-api

Il seguente esempio di codice mostra come utilizzareupdate-api.

### AWS CLI

CORSPer abilitare un HTTP API

L'update-api esempio seguente aggiorna API la CORS configurazione specificata per consentire le richieste dahttps://www.example.com.

```
aws apigatewayv2 update-api \  
  --api-id a1b2c3d4 \  
  --stage dev \  
  --domain-name api.example.com \  
  --api-mapping-id 0qzs2sy7bh \  
  --api-mapping-key dev
```

```
--api-id a1b2c3d4 \  
--cors-configuration AllowOrigins=https://www.example.com
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Configuring CORS for an HTTP API](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [UpdateApi AWS CLI](#) Command Reference.

## update-authorizer

Il seguente esempio di codice mostra come utilizzare `update-authorizer`.

### AWS CLI

Per aggiornare un autorizzatore

L'update-authorizer esempio seguente modifica l'origine dell'identità di un JWT autorizzatore in un'intestazione denominata. Authorization

```
aws apigatewayv2 update-authorizer \  
  --api-id a1b2c3d4 \  
  --authorizer-id a1b2c3 \  
  --identity-source '$request.header.Authorization'
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Controlling access to HTTP APIs with JWT authorizers](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [UpdateAuthorizer AWS CLI Command Reference](#).

## update-deployment

Il seguente esempio di codice mostra come utilizzare update-deployment.

### AWS CLI

Per modificare la descrizione di una distribuzione

L'update-deployment esempio seguente aggiorna la descrizione di una distribuzione.

```
aws apigatewayv2 update-deployment \  
  --api-id a1b2c3d4 \  
  --deployment-id a1b2c3d4 \  
  --description my-api
```

```
--api-id a1b2c3d4 \  
--deployment-id abcdef \  
--description 'Manual deployment to fix integration test failures.'
```

Output:

```
{  
  "AutoDeployed": false,  
  "CreateDate": "2020-02-05T16:21:48+00:00",  
  "DeploymentId": "abcdef",  
  "DeploymentStatus": "DEPLOYED",  
  "Description": "Manual deployment to fix integration test failures."  
}
```

Per ulteriori informazioni, consulta [Developing an HTTP API in API Gateway](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [UpdateDeployment AWS CLI Command Reference](#).

## update-domain-name

Il seguente esempio di codice mostra come utilizzare `update-domain-name`.

### AWS CLI

Per aggiornare un nome di dominio personalizzato

L'esempio seguente specifica un nuovo ACM certificato per il nome di dominio `api.example.com` personalizzato.

```
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
```

Output:

```
{  
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Configurazione di un nome di dominio personalizzato regionale in API Gateway nella](#) Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [UpdateDomainName AWS CLI](#) Command Reference.

## update-integration

Il seguente esempio di codice mostra come utilizzare `update-integration`.

### AWS CLI

Per aggiornare un'integrazione Lambda

L'`update-integration` esempio seguente aggiorna un'integrazione AWS Lambda esistente per utilizzare la funzione Lambda specificata.

```

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

```

Output:

```

{
  "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
}
```

Per ulteriori informazioni, consulta [Configurazione delle integrazioni HTTP APIs](#) e [Configurazione delle WebSocket API integrazioni](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta Command [UpdateIntegration](#) Reference AWS CLI .

## update-route

Il seguente esempio di codice mostra come utilizzare `update-route`.

### AWS CLI

Esempio 1: aggiornare l'integrazione di una rotta

L'`update-route` esempio seguente aggiorna l'integrazione di una rotta specificata.

```
aws apigatewayv2 update-route \
  --api-id a1b2c3d4 \
  --route-id a1b2c3 \
  --target integrations/a1b2c6
```

Output:

```
{
  "ApiKeyRequired": false,
  "AuthorizationType": "NONE",
  "RouteId": "a1b2c3",
  "RouteKey": "ANY /pets",
  "Target": "integrations/a1b2c6"
}
```

Esempio 2: aggiungere un autorizzatore a una rotta

L'`update-route` esempio seguente aggiorna il percorso specificato per utilizzare un JWT autorizzatore.

```
aws apigatewayv2 update-route \
  --api-id a1b2c3d4 \
  --route-id a1b2c3 \
  --authorization-type JWT \
  --authorizer-id a1b2c5 \
  --authorization-scopes user.id user.email
```

Output:

```
{
  "ApiKeyRequired": false,
  "AuthorizationScopes": [
    "user.id",
    "user.email"
  ],
  "AuthorizationType": "JWT",
  "AuthorizerId": "a1b2c5",
  "OperationName": "GET HTTP",
  "RequestParameters": {},
  "RouteId": "a1b2c3",
  "RouteKey": "GET /pets",
  "Target": "integrations/a1b2c6"
}
```

Per ulteriori informazioni, consulta [Controlling access to HTTP APIs with JWT authorizers](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [UpdateRoute AWS CLI](#) Command Reference.

## update-stage

Il seguente esempio di codice mostra come utilizzare `update-stage`.

### AWS CLI

Per configurare un throttling personalizzato

L'update-stageesempio seguente configura la limitazione personalizzata per la fase e il percorso specificati di un. API

```
aws apigatewayv2 update-stage \
  --api-id a1b2c3d4 \
```



```
--stage-name dev \  
--route-settings '{"GET /pets":  
{"ThrottlingBurstLimit":100,"ThrottlingRateLimit":2000}}'
```

Output:

```
{  
  "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": {}  
}
```

Per ulteriori informazioni, consulta [Protecting your HTTP API](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [UpdateStage AWS CLI](#) Command Reference.

## update-vpc-link

Il seguente esempio di codice mostra come utilizzare `update-vpc-link`.

### AWS CLI

Per aggiornare un VPC collegamento

L'`update-vpc-link` esempio seguente aggiorna il nome di un VPC link. Dopo aver creato un VPC link, non è possibile modificarne i gruppi di sicurezza o le sottoreti.

```
aws apigatewayv2 update-vpc-link \  
--vpc-link-id abcd123 \  

```

```
--name MyUpdatedVpcLink
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Working with VPC links for HTTP APIs](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [UpdateVpcLink AWS CLI Command Reference](#).

## API API Esempi di gestione del gateway utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with API Gateway Management API.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

## Azioni

### **delete-connection**

Il seguente esempio di codice mostra come utilizzare `delete-connection`.

#### AWS CLI

Per eliminare una WebSocket connessione

L'`delete-connection` esempio seguente disconnette un client dall'area specificata WebSocket API.

```
aws apigatewaymanagementapi delete-connection \  
  --connection-id L0SM9c0FvHcCIhw= \  
  --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Usa i comandi @connections nel tuo servizio di backend](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [DeleteConnection AWS CLI Command Reference](#).

### **get-connection**

Il seguente esempio di codice mostra come utilizzare `get-connection`.

#### AWS CLI

Per ottenere informazioni su una WebSocket connessione

L'`get-connection` esempio seguente descrive una connessione al valore specificato WebSocket API.

```
aws apigatewaymanagementapi get-connection \  
  --connection-id L0SM9c0FvHcCIhw= \  
  --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

Output:

```
{  
  "ConnectedAt": "2020-04-30T20:10:33.236Z",
```

```
"Identity": {
  "SourceIp": "192.0.2.1"
},
"LastActiveAt": "2020-04-30T20:10:42.997Z"
}
```

Per ulteriori informazioni, consulta [Usa i comandi @connections nel tuo servizio di backend](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [GetConnection AWS CLI](#) Command Reference.

## post-to-connection

Il seguente esempio di codice mostra come utilizzare `post-to-connection`.

### AWS CLI

Per inviare dati a una WebSocket connessione

L'`post-to-connection` esempio seguente invia un messaggio a un client connesso alla rete specificata 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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Usa i comandi @connections nel tuo servizio di backend](#) nella Amazon API Gateway Developer Guide.

- Per API i dettagli, consulta [PostToConnection AWS CLI](#) Command Reference.

## Esempi di App Mesh utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with App Mesh.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **create-mesh**

Il seguente esempio di codice mostra come utilizzare `create-mesh`.

#### AWS CLI

Esempio 1: Per creare una nuova rete di servizi

L'`create-mesh`esempio seguente crea una service mesh.

```
aws appmesh create-mesh \  
  --mesh-name app1
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Esempio 2: creare una nuova service mesh con più tag

L'`create-mesh` seguente crea una service mesh con più tag.

```
aws appmesh create-mesh \  
  --mesh-name app2 \  
  --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Service Meshes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [CreateMesh](#) in AWS CLI Command Reference.

## create-route

Il seguente esempio di codice mostra come utilizzare `create-route`.

### AWS CLI

Per creare una nuova RPC rotta g

L'`create-route` seguente utilizza un file JSON di input per creare un RPC percorso g. GRPC il traffico con metadati che iniziano con 123 viene indirizzato a un nodo virtuale denominato. serviceBgrpc Se si verificano TCP errori specifici g RPC HTTP, o durante il tentativo di comunicare con la destinazione della route, la route viene ritentata tre volte. C'è un ritardo di 15 secondi tra ogni tentativo.

```
aws appmesh create-route \  
--cli-input-json file://create-route-grpc.json
```

Contenuto di 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"
  }

```

### Output:

```

{
  "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"
}
}

```

Per creare una nuova rotta HTTP o HTTP /2

L'`create-route` esempio seguente utilizza un file JSON di input per creare una route HTTP /2. Per creare una HTTP rotta, sostituisci `Http2Route` con `httpRoute` under `spec`. Tutto il traffico HTTP /2 indirizzato a qualsiasi URL prefisso con un valore di intestazione che inizia con 123 viene indirizzato a un nodo virtuale denominato `2. serviceBhttp`. Se si verificano errori specifici HTTP o si TCP verificano errori nel tentativo di comunicare con la destinazione del percorso, il percorso viene ritentato tre volte. C'è un ritardo di 15 secondi tra ogni tentativo.

```

aws appmesh create-route \
  --cli-input-json file://create-route-http2.json

```

Contenuto di `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"
}
```

## Output:

```
{
  "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"
}
}

```

Per creare un nuovo percorso TCP

L'`create-route` seguente utilizza un file JSON di input per creare un TCP percorso. Il 75 per cento del traffico viene indirizzato a un nodo virtuale denominato `serviceBtcp` e il 25 per cento del traffico viene instradato a un nodo virtuale denominato `serviceBv2tcp`. Specificare ponderazioni diverse per obiettivi diversi è un modo efficace per eseguire la distribuzione di una nuova versione di un'applicazione. È possibile modificare i pesi in modo che alla fine il 100% di tutto il traffico venga indirizzato a una destinazione che dispone della nuova versione di un'applicazione.

```

aws appmesh create-route \
  --cli-input-json file://create-route-tcp.json

```

Contenuto di `create-route-tcp.json`:

```

{
  "meshName": "apps",
  "routeName": "tcpRoute",
  "spec": {
    "priority": 300,
    "tcpRoute": {
      "action": {
        "weightedTargets": [
          {

```

```

        "virtualNode": "serviceBtcp",
        "weight": 75
      },
      {
        "virtualNode": "serviceBv2tcp",
        "weight": 25
      }
    ]
  }
},
"virtualRouterName": "serviceBtcp"
}

```

### Output:

```

{
  "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"
}
}

```

Per ulteriori informazioni, consulta [Routes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [CreateRoute](#) in AWS CLI Command Reference.

## create-virtual-gateway

Il seguente esempio di codice mostra come utilizzare `create-virtual-gateway`.

### AWS CLI

Per creare un nuovo gateway virtuale

L'`create-virtual-gateway` esempio seguente utilizza un file JSON di input per creare un gateway virtuale con un listener per HTTP l'utilizzo della porta 9080.

```

aws appmesh create-virtual-gateway \
  --mesh-name meshName \
  --virtual-gateway-name virtualGatewayName \
  --cli-input-json file://create-virtual-gateway.json

```

Contenuto di `create-virtual-gateway.json`.

```

{
  "spec": {
    "listeners": [
      {
        "portMapping": {
          "port": 9080,
          "protocol": "http"
        }
      }
    ]
  }
}

```

```
}
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Virtual Gateways](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [CreateVirtualGateway](#) in AWS CLI Command Reference.

## create-virtual-node

Il seguente esempio di codice mostra come utilizzare `create-virtual-node`.

## AWS CLI

Esempio 1: per creare un nuovo nodo virtuale da utilizzare DNS per il rilevamento

L'`create-virtual-node` seguente utilizza un file JSON di input per creare un nodo virtuale utilizzato DNS per l'individuazione dei servizi.

```
aws appmesh create-virtual-node \  
  --cli-input-json file://create-virtual-node-dns.json
```

Contenuto di `create-virtual-node-dns.json`.

```
{  
  "meshName": "app1",  
  "spec": {  
    "listeners": [  
      {  
        "portMapping": {  
          "port": 80,  
          "protocol": "http"  
        }  
      }  
    ],  
    "serviceDiscovery": {  
      "dns": {  
        "hostname": "serviceBv1.svc.cluster.local"  
      }  
    }  
  },  
  "virtualNodeName": "vnServiceBv1"  
}
```

Output:

```
{  
  "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"
}
}

```

Esempio 2: creare un nuovo nodo virtuale che utilizza AWS Cloud Map per il rilevamento

L'create-virtual-nodeesempio seguente utilizza un file JSON di input per creare un nodo virtuale che utilizza AWS Cloud Map per l'individuazione dei servizi.

```

aws appmesh create-virtual-node \
  --cli-input-json file://create-virtual-node-cloud-map.json

```

Contenuto di 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"
}

```

### Output:

```

{
  "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"
}
}
```

Per ulteriori informazioni, consulta [Virtual Nodes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [CreateVirtualNode](#) in AWS CLI Command Reference.

## **create-virtual-router**

Il seguente esempio di codice mostra come utilizzare `create-virtual-router`.

### AWS CLI

Per creare un nuovo router virtuale

L'create-virtual-router esempio seguente utilizza un file JSON di input per creare un router virtuale con un listener per HTTP l'utilizzo della porta 80.

```
aws appmesh create-virtual-router \  
--cli-input-json file://create-virtual-router.json
```

Contenuto di create-virtual-router.json.

```
{  
  "meshName": "app1",  
  "spec": {  
    "listeners": [  
      {  
        "portMapping": {  
          "port": 80,  
          "protocol": "http"  
        }  
      }  
    ]  
  },  
  "virtualRouterName": "vrServiceB"  
}
```

Output:

```
{  
  "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"
}
}

```

Per ulteriori informazioni, consulta [Virtual Router](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [CreateVirtualRouter](#) in AWS CLI Command Reference.

## create-virtual-service

Il seguente esempio di codice mostra come utilizzare `create-virtual-service`.

### AWS CLI

Esempio 1: creare un nuovo servizio virtuale con un provider di nodi virtuali

L'esempio seguente utilizza un file JSON di input per creare un servizio virtuale con un provider di nodi virtuali.

```
aws appmesh create-virtual-service \
  --cli-input-json file://create-virtual-service-virtual-node.json
```

Contenuto di `create-virtual-service-virtual-node.json`.

```

{
  "meshName": "app1",
  "spec": {
    "provider": {
      "virtualNode": {
        "virtualNodeName": "vnServiceA"
      }
    }
  },
  "virtualServiceName": "serviceA.svc.cluster.local"
}

```

**Output:**

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Virtual Node](#) nella AWS App Mesh User Guide.

Esempio 2: creare un nuovo servizio virtuale con un provider di router virtuale

L'`create-virtual-serviceesempio` seguente utilizza un file JSON di input per creare un servizio virtuale con un provider di router virtuale.

```
aws appmesh create-virtual-service \
  --cli-input-json file://create-virtual-service-virtual-router.json
```

Contenuto di `create-virtual-service-virtual-router.json`.

```
{
  "meshName": "app1",
  "spec": {
    "provider": {
```

```

        "virtualRouter": {
            "virtualRouterName": "vrServiceB"
        }
    },
    "virtualServiceName": "serviceB.svc.cluster.local"
}

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta 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) > nella App Mesh User Guide AWS

- Per API i dettagli, vedere in Command Reference. [CreateVirtualService](#) AWS CLI

## delete-mesh

Il seguente esempio di codice mostra come utilizzare `delete-mesh`.

## AWS CLI

Per eliminare una service mesh

L'delete-meshesempio seguente elimina la service mesh specificata.

```
aws appmesh delete-mesh \  
  --mesh-name app1
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Service Meshes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [DeleteMesh](#) in AWS CLI Command Reference.

## delete-route

Il seguente esempio di codice mostra come utilizzare delete-route.

## AWS CLI

Per eliminare un percorso



L'`delete-route` seguente elimina la rotta specificata.

```
aws appmesh delete-route \  
  --mesh-name app1 \  
  --virtual-router-name vrServiceB \  
  --route-name toVnServiceB-weighted
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Routes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [DeleteRoute](#) in AWS CLI Command Reference.

## delete-virtual-node

Il seguente esempio di codice mostra come utilizzare `delete-virtual-node`.

### AWS CLI

Per eliminare un nodo virtuale

L'`delete-virtual-node` esempio seguente elimina il nodo virtuale specificato.

```

aws appmesh delete-virtual-node \
  --mesh-name app1 \
  --virtual-node-name vnServiceBv2

```

Output:

```

{
  "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"
}
}

```

Per ulteriori informazioni, consulta [Virtual Nodes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [DeleteVirtualNode](#) in AWS CLI Command Reference.

## delete-virtual-router

Il seguente esempio di codice mostra come utilizzare `delete-virtual-router`.

### AWS CLI

Per eliminare un router virtuale

L'`delete-virtual-router` esempio seguente elimina il router virtuale specificato.

```

aws appmesh delete-virtual-router \
  --mesh-name app1 \
  --virtual-router-name vrServiceB

```

Output:

```

{
  "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"
}
}

```

Per ulteriori informazioni, consulta [Virtual Router](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [DeleteVirtualRouter](#) in AWS CLI Command Reference.

## delete-virtual-service

Il seguente esempio di codice mostra come utilizzare `delete-virtual-service`.

### AWS CLI

Per eliminare un servizio virtuale

L'`delete-virtual-service` esempio seguente elimina il servizio virtuale specificato.

```

aws appmesh delete-virtual-service \
  --mesh-name app1 \
  --virtual-service-name serviceB.svc.cluster.local

```

Output:

```

{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Virtual Service](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [DeleteVirtualService](#) in AWS CLI Command Reference.

## describe-mesh

Il seguente esempio di codice mostra come utilizzare `describe-mesh`.

### AWS CLI

Per descrivere una service mesh

L'`describe-mesh` esempio seguente restituisce i dettagli sulla service mesh specificata.

```
aws appmesh describe-mesh \
  --mesh-name app1
```

Output:

```
{
  "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"
    }
}
}

```

Per ulteriori informazioni, consulta [Service Meshes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [DescribeMesh](#) in AWS CLI Command Reference.

## describe-route

Il seguente esempio di codice mostra come utilizzare `describe-route`.

### AWS CLI

Per descrivere un percorso

L'`describe-route` esempio seguente restituisce i dettagli sulla rotta specificata.

```

aws appmesh describe-route \
  --mesh-name app1 \
  --virtual-router-name vrServiceB \
  --route-name toVnServiceB-weighted

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Routes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [DescribeRoute](#) in AWS CLI Command Reference.

## describe-virtual-node

Il seguente esempio di codice mostra come utilizzare `describe-virtual-node`.

### AWS CLI

Per descrivere un nodo virtuale

L'`describe-virtual-node` esempio seguente restituisce i dettagli sul nodo virtuale specificato.

```

aws appmesh describe-virtual-node \
  --mesh-name app1 \
  --virtual-node-name vnServiceBv1

```

**Output:**

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Virtual Nodes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [DescribeVirtualNode](#) in AWS CLI Command Reference.

**describe-virtual-router**

Il seguente esempio di codice mostra come utilizzare `describe-virtual-router`.



## AWS CLI

Per descrivere un router virtuale

L'`describe-virtual-router` seguente restituisce i dettagli sul router virtuale specificato.

```
aws appmesh describe-virtual-router \
  --mesh-name app1 \
  --virtual-router-name vrServiceB
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Virtual Router](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [DescribeVirtualRouter](#) in AWS CLI Command Reference.

## describe-virtual-service

Il seguente esempio di codice mostra come utilizzare `describe-virtual-service`.

### AWS CLI

Per descrivere un servizio virtuale

L'`describe-virtual-service` esempio seguente restituisce i dettagli sul servizio virtuale specificato.

```
aws appmesh describe-virtual-service \  
  --mesh-name app1 \  
  --virtual-service-name serviceB.svc.cluster.local
```

Output:

```
{  
  "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"  
  }  
}
```

```
}
```

Per ulteriori informazioni, consulta [Virtual Services](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [DescribeVirtualService](#) in AWS CLI Command Reference.

## list-meshes

Il seguente esempio di codice mostra come utilizzare `list-meshes`.

### AWS CLI

Per elencare le mesh di servizio

L'`list-meshes` esempio seguente elenca tutte le mesh di servizio nella regione corrente AWS .

```
aws appmesh list-meshes
```

Output:

```
{
  "meshes": [
    {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
      "meshName": "app1"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Service Meshes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [ListMeshes](#) in AWS CLI Command Reference.

## list-routes

Il seguente esempio di codice mostra come utilizzare `list-routes`.

### AWS CLI

Per elencare i percorsi

L'`list-routes` esempio seguente elenca tutte le rotte per il router virtuale specificato.

```
aws appmesh list-routes \  
  --mesh-name app1 \  
  --virtual-router-name vrServiceB
```

Output:

```
{  
  "routes": [  
    {  
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/  
vrServiceB/route/toVnServiceB",  
      "meshName": "app1",  
      "routeName": "toVnServiceB-weighted",  
      "virtualRouterName": "vrServiceB"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Routes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [ListRoutes](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

AWS CLI

Per elencare i tag di una risorsa

L'`list-tags-for-resource` esempio seguente elenca tutti i tag assegnati alla risorsa specificata.

```
aws appmesh list-tags-for-resource \  
  --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1
```

Output:

```
{  
  "tags": [  

```

```
{
  "key": "key1",
  "value": "value1"
},
{
  "key": "key2",
  "value": "value2"
},
{
  "key": "key3",
  "value": "value3"
}
]
```

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## list-virtual-nodes

Il seguente esempio di codice mostra come utilizzare `list-virtual-nodes`.

### AWS CLI

Per elencare i nodi virtuali

L'`list-virtual-nodes` esempio seguente elenca tutti i nodi virtuali nella mesh di servizio specificata.

```
aws appmesh list-virtual-nodes \
  --mesh-name app1
```

Output:

```
{
  "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"
}
]
```

Per ulteriori informazioni, consulta [Virtual Nodes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [ListVirtualNodes](#) in AWS CLI Command Reference.

## list-virtual-routers

Il seguente esempio di codice mostra come utilizzare `list-virtual-routers`.

### AWS CLI

Per elencare i router virtuali

L'`list-virtual-routers` esempio seguente elenca tutti i router virtuali nella service mesh specificata.

```
aws appmesh list-virtual-routers \
  --mesh-name app1
```

Output:

```
{
  "virtualRouters": [
    {
      "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
      "meshName": "app1",
      "virtualRouterName": "vrServiceB"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Virtual Router](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [ListVirtualRouters](#) in AWS CLI Command Reference.

## list-virtual-services

Il seguente esempio di codice mostra come utilizzare `list-virtual-services`.

### AWS CLI

Per elencare i servizi virtuali

L'`list-virtual-services` esempio seguente elenca tutti i servizi virtuali nella mesh di servizi specificata.

```
aws appmesh list-virtual-services \
  --mesh-name app1
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Virtual Services](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [ListVirtualServices](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

## AWS CLI

Per etichettare una risorsa

L'`tag-resource` seguente aggiunge il tag `key1` con il valore `value1` alla risorsa specificata.

```
aws appmesh tag-resource \  
  --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \  
  --tags key=key1,value=value1
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

## AWS CLI

Per rimuovere i tag da una risorsa

L'`untag-resource` seguente rimuove un tag con la chiave `key1` dalla risorsa specificata.

```
aws appmesh untag-resource \  
  --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \  
  --tag-keys key1
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-mesh

Il seguente esempio di codice mostra come utilizzare `update-mesh`.

## AWS CLI

Per aggiornare una service mesh



L'update-meshesempio seguente utilizza un file JSON di input per aggiornare una service mesh per consentire l'inoltro inalterato di tutto il traffico esterno in uscita attraverso il proxy Envoy.

```
aws appmesh update-mesh \  
  --cli-input-json file://update-mesh.json
```

Contenuto di update-mesh.json.

```
{  
  "meshName": "app1",  
  "spec": {  
    "egressFilter": {  
      "type": "ALLOW_ALL"  
    }  
  }  
}
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Service Meshes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [UpdateMesh](#) in AWS CLI Command Reference.

## update-route

Il seguente esempio di codice mostra come utilizzare `update-route`.

### AWS CLI

Per aggiornare un percorso

L'`update-route` esempio seguente utilizza un file JSON di input per aggiornare i pesi di un percorso.

```
aws appmesh update-route \  
  --cli-input-json file://update-route-weighted.json
```

Contenuto di `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"  
}
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Routes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [UpdateRoute](#) in AWS CLI Command Reference.

## update-virtual-node

Il seguente esempio di codice mostra come utilizzare `update-virtual-node`.

### AWS CLI

Per aggiornare un nodo virtuale

L'`update-virtual-node`esempio seguente utilizza un file JSON di input per aggiungere un controllo dello stato di salute a un nodo virtuale.

```
aws appmesh update-virtual-node \  
  --cli-input-json file://update-virtual-node.json
```

Contenuto di `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"  
}
```

## Output:

```
{  
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Virtual Nodes](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [UpdateVirtualNode](#) in AWS CLI Command Reference.

## update-virtual-router

Il seguente esempio di codice mostra come utilizzare `update-virtual-router`.

### AWS CLI

Per aggiornare un router virtuale

L'`update-virtual-router` esempio seguente utilizza un file JSON di input per aggiornare una porta listener del router virtuale.

```
aws appmesh update-virtual-router \
  --cli-input-json file://update-virtual-router.json
```

Contenuto di `update-virtual-router.json`.

```
{
  "meshName": "app1",
  "spec": {
    "listeners": [
      {
        "portMapping": {
          "port": 8080,
          "protocol": "http"
        }
      }
    ]
  },
  "virtualRouterName": "vrServiceB"
}
```

Output:

```
{
```

```
"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"
}
}
```

Per ulteriori informazioni, consulta [Virtual Router](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [UpdateVirtualRouter](#) in AWS CLI Command Reference.

## update-virtual-service

Il seguente esempio di codice mostra come utilizzare `update-virtual-service`.

### AWS CLI

Per aggiornare un servizio virtuale

L'esempio seguente utilizza un file JSON di input per aggiornare un servizio virtuale per utilizzare un provider di router virtuale.

```
aws appmesh update-virtual-service \
```

```
--cli-input-json file://update-virtual-service.json
```

Contenuto di `update-virtual-service.json`.

```
{
  "meshName": "app1",
  "spec": {
    "provider": {
      "virtualRouter": {
        "virtualRouterName": "vrServiceA"
      }
    }
  },
  "virtualServiceName": "serviceA.svc.cluster.local"
}
```

Output:

```
{
  "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"
  }
}
```



Per ulteriori informazioni, consulta [Virtual Services](#) nella AWS App Mesh User Guide.

- Per API i dettagli, vedere [UpdateVirtualService](#) in AWS CLI Command Reference.

## Esempi di App Runner utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface con App Runner.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **associate-custom-domain**

Il seguente esempio di codice mostra come utilizzare `associate-custom-domain`.

#### AWS CLI

Per associare un nome di dominio e il sottodominio `www` a un servizio

L'`associate-custom-domain` esempio seguente associa un nome di dominio personalizzato controllato dall'utente a un servizio App Runner. Il nome di dominio è il dominio principale `example.com`, incluso il sottodominio per casi speciali. `www.example.com`

```
aws apprunner associate-custom-domain \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",
```

```
"DomainName": "example.com",
"EnableWWWSubdomain": true
}
```

## Output:

```
{
  "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.mzkbsghbfwj.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"
}
```

- Per API i dettagli, vedere [AssociateCustomDomain](#) in AWS CLI Command Reference.

## create-auto-scaling-configuration

Il seguente esempio di codice mostra come utilizzare `create-auto-scaling-configuration`.

### AWS CLI

Per creare una configurazione di scalabilità automatica ad alta disponibilità

L'create-auto-scaling-configurationesempio seguente crea una configurazione di scalabilità automatica ottimizzata per l'alta disponibilità impostando MinSize su 5. Con questa configurazione, App Runner tenta di distribuire le istanze del servizio nella maggior parte delle zone di disponibilità possibili, fino a cinque, a seconda della regione. AWS

La chiamata restituisce un AutoScalingConfiguration oggetto con le altre impostazioni impostate sui valori predefiniti. Nell'esempio, questa è la prima chiamata per creare una configurazione denominata. high-availability La revisione è impostata su 1 ed è la revisione più recente.

```
aws apprunner create-auto-scaling-configuration \  
  --cli-input-json file://input.json
```

Contenuto di input.json.

```
{  
  "AutoScalingConfigurationName": "high-availability",  
  "MinSize": 5  
}
```

Output:

```
{  
  "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  
  }  
}
```

- Per API i dettagli, vedere [CreateAutoScalingConfiguration](#) in AWS CLI Command Reference.

## create-connection

Il seguente esempio di codice mostra come utilizzare `create-connection`.

### AWS CLI

Per creare una GitHub connessione

L'`create-connection` seguente crea una connessione a un archivio di GitHub codice privato. Lo stato della connessione dopo una chiamata riuscita è `PENDING_HANDSHAKE`. Questo perché non è ancora avvenuta una stretta di autenticazione con il provider. Completa l'`handshake` utilizzando la console App Runner.

```
aws apprunner create-connection \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "ConnectionName": "my-github-connection",  
  "ProviderType": "GITHUB"  
}
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Gestire le connessioni App Runner](#) nella App Runner Developer AWS Guide.

- Per API i dettagli, vedere [CreateConnection](#) in AWS CLI Command Reference.

## create-service

Il seguente esempio di codice mostra come utilizzare `create-service`.

### AWS CLI

Esempio 1: creare un servizio di deposito del codice sorgente

L'esempio seguente crea un servizio App Runner basato su un repository di codice sorgente Python.

```
aws apprunner create-service \  
--cli-input-json file://input.json
```

Contenuto di `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"
  }
}

```

### Output:

```

{
  "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"
    }
}
}

```

## Esempio 2: creare un servizio di archiviazione del codice sorgente

L'`create-service` seguente crea un servizio App Runner basato su un repository di codice sorgente Python.

```

aws apprunner create-service \
  --cli-input-json file://input.json

```

Contenuto di `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"
}
}

```

### Output:

```

{
  "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"
}
}
}
}

```

### Esempio 3: creare un servizio di archivio di immagini di origine

L'`create-service` seguente crea un servizio App Runner basato su un'immagine memorizzata in Elastic Container Registry (ECR).

```

aws apprunner create-service \
  --cli-input-json file://input.json

```

Contenuto di `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"
}
}

```

### Output:

```

{
  "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"
}
}
}

```

- Per API i dettagli, consulta [CreateService AWS CLI Command Reference](#).

## delete-auto-scaling-configuration

Il seguente esempio di codice mostra come utilizzare `delete-auto-scaling-configuration`.

### AWS CLI

Esempio 1: eliminare l'ultima revisione attiva di una configurazione di autoscaling

L'`delete-auto-scaling-configuration` seguente elimina l'ultima revisione attiva di una configurazione di scalabilità automatica di App Runner. Per eliminare l'ultima revisione attiva, specifica un Amazon Resource Name (ARN) che termina con il nome della configurazione, senza il componente di revisione.

Nell'esempio, esistono due revisioni prima di questa azione. Pertanto, la revisione 2 (l'ultima) viene eliminata. Tuttavia, ora viene visualizzata `"Latest": false`, perché, dopo essere stata eliminata, non è più l'ultima revisione attiva.

```
aws apprunner delete-auto-scaling-configuration \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability"
```

```
}
```

Output:

```
{
  "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
  }
}
```

Esempio 2: eliminare una revisione specifica di una configurazione di ridimensionamento automatico

L'`delete-auto-scaling-configuration` seguente elimina una revisione specifica di una configurazione di scalabilità automatica di App Runner. Per eliminare una revisione specifica, specificare una ARN che includa il numero di revisione.

Nell'esempio, esistono diverse revisioni prima di questa azione. L'azione elimina la revisione. 1

```
aws apprunner delete-auto-scaling-configuration \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/1"
}
```

Output:

```
{
  "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
  }
}
```

- Per API i dettagli, vedere [DeleteAutoScalingConfiguration](#) in AWS CLI Command Reference.

## delete-connection

Il seguente esempio di codice mostra come utilizzare `delete-connection`.

### AWS CLI

Per eliminare una connessione

L'`delete-connection` esempio seguente elimina una connessione App Runner. Lo stato della connessione dopo una chiamata riuscita è `DELETED`. Questo perché la connessione non è più disponibile.

```
aws apprunner delete-connection \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-github-
connection"
}
```

**Output:**

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [DeleteConnection](#) in AWS CLI Command Reference.

**delete-service**

Il seguente esempio di codice mostra come utilizzare `delete-service`.

**AWS CLI**

Per eliminare un servizio

L'`delete-service` esempio seguente elimina un servizio App Runner.

```
aws apprunner delete-service \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

**Output:**

```
{
  "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"
    }
  }
}
```

- Per API i dettagli, vedere [DeleteService](#) in AWS CLI Command Reference.

## describe-auto-scaling-configuration

Il seguente esempio di codice mostra come utilizzare `describe-auto-scaling-configuration`.

### AWS CLI

Esempio 1: Per descrivere l'ultima revisione attiva di una configurazione di auto scaling

L'esempio seguente ottiene una descrizione dell'ultima revisione attiva di una configurazione di scalabilità automatica di App Runner. Per descrivere l'ultima revisione attiva, specifica una ARN che termina con il nome della configurazione, senza il componente di revisione.

Nell'esempio, esistono due revisioni. Pertanto, viene descritta la revisione 2 (la più recente). L'oggetto risultante mostra `"Latest": true`.

```
aws apprunner describe-auto-scaling-configuration \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-  
east-1:123456789012:autoscalingconfiguration/high-availability"  
}
```

Output:

```
{  
  "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
```



```
}
}
```

Esempio 2: descrivere una revisione specifica di una configurazione di autoscaling

L'`describe-auto-scaling-configuration`esempio seguente ottiene una descrizione di una revisione specifica di una configurazione di scalabilità automatica di App Runner. Per descrivere una revisione specifica, specificane una ARN che includa il numero di revisione.

Nell'esempio, esistono diverse revisioni e viene richiesta la revisione1. L'oggetto risultante mostra. "Latest": false

```
aws apprunner describe-auto-scaling-configuration \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/1"
}
```

Output:

```
{
  "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
  }
}
```

- Per API i dettagli, vedere [DescribeAutoScalingConfiguration](#) in AWS CLI Command Reference.

## describe-custom-domains

Il seguente esempio di codice mostra come utilizzare `describe-custom-domains`.

### AWS CLI

Per ottenere descrizioni dei nomi di dominio personalizzati associati a un servizio

L'esempio seguente ottiene le descrizioni e lo stato dei nomi di dominio personalizzati associati a un servizio App Runner.

```
aws apprunner describe-custom-domains \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
  "DomainName": "example.com",
  "EnableWWWSubdomain": true
}
```

Output:

```
{
  "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"
}

```

- Per API i dettagli, vedere [DescribeCustomDomains](#) in AWS CLI Command Reference.

## describe-service

Il seguente esempio di codice mostra come utilizzare `describe-service`.

### AWS CLI

Per descrivere un servizio

L'esempio seguente ottiene una descrizione di un servizio App Runner.

```
aws apprunner describe-service \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

## Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [DescribeService](#) in AWS CLI Command Reference.

## disassociate-custom-domain

Il seguente esempio di codice mostra come utilizzare `disassociate-custom-domain`.

### AWS CLI

Per dissociare un nome di dominio da un servizio

L'esempio seguente dissocia il dominio `example.com` da un servizio App Runner. La chiamata dissocia inoltre il sottodominio `www.example.com` associato al dominio radice.

```
aws apprunner disassociate-custom-domain \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
  "DomainName": "example.com"
}
```

Output:

```
{
  "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": "DELETING"
},
"DNSTarget": "psbqam834h.us-east-1.awsapprunner.com",
"ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}

```

- Per API i dettagli, vedere [DisassociateCustomDomain](#) in AWS CLI Command Reference.

## list-auto-scaling-configurations

Il seguente esempio di codice mostra come utilizzare `list-auto-scaling-configurations`.

### AWS CLI

Per ottenere un elenco impaginato delle configurazioni di autoscaling di App Runner

L'`list-auto-scaling-configurations` esempio seguente elenca tutte le configurazioni di scalabilità automatica di App Runner presenti nell'account. AWS In ogni risposta sono elencate fino a cinque configurazioni di auto scaling. `AutoScalingConfigurationName` e `LatestOnly` non sono specificate. I loro valori predefiniti fanno sì che venga elencata la revisione più recente di tutte le configurazioni attive.

In questo esempio, la risposta include due risultati e non ce ne sono altri, quindi viene restituito `no. NextToken`

```
aws apprunner list-auto-scaling-configurations \
```

```
--cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "MaxResults": 5
}
```

Output:

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [ListAutoScalingConfigurations](#) in AWS CLI Command Reference.

## list-connections

Il seguente esempio di codice mostra come utilizzare `list-connections`.

### AWS CLI

Esempio 1: Per elencare tutte le connessioni

L'`list-connections` esempio seguente elenca tutte le connessioni App Runner nell' AWS account.

```
aws apprunner list-connections
```

Output:

```
{
  "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"
    }
  ]
}
```

Esempio 2: per elencare una connessione per nome

L'`list-connections` esempio seguente elenca una connessione in base al nome.

```
aws apprunner list-connections \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "ConnectionName": "my-github-org-connection"
}
```

Output:

```
{
```



```
"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"  
  }  
]
```

- Per API i dettagli, vedere [ListConnections](#) in AWS CLI Command Reference.

## list-operations

Il seguente esempio di codice mostra come utilizzare `list-operations`.

### AWS CLI

Per elencare le operazioni avvenute su un servizio

L'`list-operations` seguente elenca tutte le operazioni eseguite finora su un servizio App Runner. In questo esempio, il servizio è nuovo e si è verificata una sola operazione di tipo `CREATE_SERVICE`.

```
aws apprunner list-operations \  
--cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa"  
}
```

Output:

```
{  
  "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
    }
]
}

```

- Per API i dettagli, vedere [ListOperations](#) in AWS CLI Command Reference.

## list-services

Il seguente esempio di codice mostra come utilizzare `list-services`.

### AWS CLI

Per ottenere un elenco impaginato dei servizi App Runner

L'`list-services` esempio seguente elenca tutti i servizi App Runner presenti nell'account. AWS In ogni risposta sono elencati fino a due servizi. Questo esempio mostra la prima richiesta. La risposta include due risultati e un token che può essere utilizzato nella richiesta successiva. Quando una risposta successiva non include un token, tutti i servizi sono stati elencati.

```

aws apprunner list-services \
  --cli-input-json file://input.json

```

Contenuto di `input.json`.

```

{
  "MaxResults": 2
}

```

Output:

```

{
  "NextToken":
  "eyJDDdXN0b21lckFjY291bnRjZCI6IjI3MDIwNTQwMjg0NSIsI1NlcnZpY2VTdGF0dXNDb2RlIjojUFJpVkk1TSU90SU
  "ServiceSummaryList": [

```

```

    {
      "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",
      "Status": "RUNNING"
    },
    {
      "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"
    }
  ]
}

```

- Per API i dettagli, vedere [ListServices](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag associati a un servizio App Runner

L'`list-tags-for-resource` esempio seguente elenca tutti i tag associati a un servizio App Runner.

```
aws apprunner list-tags-for-resource \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
```

```
"ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

### Output:

```
{
  "Tags": [
    {
      "Key": "Department",
      "Value": "Retail"
    },
    {
      "Key": "CustomerId",
      "Value": "56439872357912"
    }
  ]
}
```

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## pause-service

Il seguente esempio di codice mostra come utilizzare `pause-service`.

### AWS CLI

Per mettere in pausa un servizio

L'esempio seguente mette in pausa un servizio App Runner.

```
aws apprunner pause-service \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

## Output:

```
{
  "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"
      }
    }
  }
}

```

- Per API i dettagli, vedere [PauseService](#) in AWS CLI Command Reference.

## resume-service

Il seguente esempio di codice mostra come utilizzare `resume-service`.

### AWS CLI

Per riprendere un servizio

L'esempio seguente riprende un servizio App Runner.

```

aws apprunner resume-service \
  --cli-input-json file://input.json

```

Contenuto di `input.json`.

```

{
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-app/8fe1e10304f84fd2b0df550fe98a71fa"
}

```

Output:

```

{
  "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"
}
}
}

```

- Per API i dettagli, vedere [ResumeService](#) in AWS CLI Command Reference.

## start-deployment

Il seguente esempio di codice mostra come utilizzare `start-deployment`.

## AWS CLI

Per avviare una distribuzione manuale

L'`start-deployment` seguente esegue una distribuzione manuale su un servizio App Runner.

```
aws apprunner start-deployment \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa"  
}
```

Output:

```
{  
  "OperationId": "853a7d5b-fc9f-4730-831b-fd8037ab832a"  
}
```

- Per API i dettagli, vedere [StartDeployment](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

## AWS CLI

Per aggiungere tag a un servizio App Runner

L'`tag-resource` seguente aggiunge due tag a un servizio App Runner.

```
aws apprunner tag-resource \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.



```
{
  "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
  "Tags": [
    {
      "Key": "Department",
      "Value": "Retail"
    },
    {
      "Key": "CustomerId",
      "Value": "56439872357912"
    }
  ]
}
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da un servizio App Runner

L'`untag-resource` esempio seguente rimuove due tag da un servizio App Runner.

```
aws apprunner untag-resource \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
  "TagKeys": [
    "Department",
    "CustomerId"
  ]
}
```

```
}
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-service

Il seguente esempio di codice mostra come utilizzare `update-service`.

### AWS CLI

Per aggiornare le dimensioni della memoria

L'update-service esempio seguente aggiorna la dimensione della memoria delle istanze (unità di scala) di un servizio App Runner a 2048 MiB.

Quando la chiamata ha esito positivo, App Runner avvia un processo di aggiornamento asincrono. La Service struttura restituita dalla chiamata riflette il nuovo valore di memoria applicato da questa chiamata.

```
aws apprunner update-service \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-  
app/8fe1e10304f84fd2b0df550fe98a71fa",  
  "InstanceConfiguration": {  
    "Memory": "4 GB"  
  }  
}
```

Output:

```
{  
  "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"
    }
  }
}

```

- Per API i dettagli, vedere [UpdateService](#) in AWS CLI Command Reference.

## AWS AppConfig esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS AppConfig.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-application**

Il seguente esempio di codice mostra come utilizzare `create-application`.

AWS CLI

Per creare un'applicazione

L'`create-application` esempio seguente crea un'applicazione in AWS AppConfig.

```
aws appconfig create-application \  
  --name "example-application" \  
  --description "An application used for creating an example."
```

Output:

```
{  
  "Description": "An application used for creating an example.",  
  "Id": "339ohji",  
  "Name": "example-application"  
}
```

Per ulteriori informazioni, vedere [Fase 1: Creazione di un' AWS AppConfig applicazione](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [CreateApplication](#) in AWS CLI Command Reference.

## create-configuration-profile

Il seguente esempio di codice mostra come utilizzare `create-configuration-profile`.

### AWS CLI

Per creare un profilo di configurazione

L'esempio seguente crea un profilo di configurazione utilizzando una configurazione memorizzata in Parameter Store, una funzionalità di Systems Manager.

```
aws appconfig create-configuration-profile \  
  --application-id "339ohji" \  
  --name "Example-Configuration-Profile" \  
  --location-uri "ssm-parameter://Example-Parameter" \  
  --retrieval-role-arn "arn:aws:iam::111122223333:role/Example-App-Config-Role"
```

Output:

```
{  
  "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  
}
```

Per ulteriori informazioni, vedere [Fase 3: Creazione di una configurazione e di un profilo di configurazione](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [CreateConfigurationProfile](#) in AWS CLI Command Reference.

## create-environment

Il seguente esempio di codice mostra come utilizzare `create-environment`.

## AWS CLI

Per creare un ambiente

L'`create-environment` seguente crea un AWS AppConfig ambiente denominato `Example-Environment` utilizzando l'applicazione creata utilizzando `create-application`.

```
aws appconfig create-environment \  
  --application-id "339ohji" \  
  --name "Example-Environment"
```

Output:

```
{  
  "ApplicationId": "339ohji",  
  "Description": null,  
  "Id": "54j1r29",  
  "Monitors": null,  
  "Name": "Example-Environment",  
  "State": "ReadyForDeployment"  
}
```

Per ulteriori informazioni, consulta [Fase 2: Creazione di un ambiente](#) nella Guida per l'utente AWS AppConfig

- Per API i dettagli, vedere [CreateEnvironment](#) in AWS CLI Command Reference.

## `create-extension-association`

Il seguente esempio di codice mostra come utilizzare `create-extension-association`.

## AWS CLI

Per creare un'associazione di estensioni

L'`create-extension-association` seguente crea una nuova associazione di estensioni in AWS AppConfig.

```
aws appconfig create-extension-association \  
  --region us-west-2 \  
  --extension-identifier S3-backup-extension \  
  --application-id 339ohji
```

```
--resource-identifier "arn:aws:appconfig:us-west-2:123456789012:application/Finance" \
--parameters S3bucket=FinanceConfigurationBackup
```

Output:

```
{
  "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
}
```

Per ulteriori informazioni, consultate [Lavorare con AWS AppConfig le estensioni](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [CreateExtensionAssociation](#) in AWS CLI Command Reference.

## create-extension

Il seguente esempio di codice mostra come utilizzare `create-extension`.

### AWS CLI

Per creare un'estensione

L'`create-extension` esempio seguente crea una nuova estensione 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}
```

Output:

```
{
  "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
    }
  }
}
```

Per ulteriori informazioni, vedete [Lavorare con AWS AppConfig le estensioni](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [CreateExtension](#) in AWS CLI Command Reference.

## create-hosted-configuration-version

Il seguente esempio di codice mostra come utilizzare `create-hosted-configuration-version`.

### AWS CLI

Per creare una versione di configurazione ospitata

L'`create-hosted-configuration-version` esempio seguente crea una nuova configurazione nell'archivio di configurazione AWS AppConfig ospitato. Il contenuto della configurazione deve essere prima convertito in base64.

```
aws appconfig create-hosted-configuration-version \
  --application-id "339ohji" \
  --configuration-profile-id "ur8hx2f" \
```



```
--
content eyAiTmFtZSI6ICJFeGFtcGxlQXBwbGljYXRpb24iLCAiSWQiOiBFFeGFtcGxlSUQsICJSYW5rIjogMyB9
\
--content-type "application/json" \
configuration_version_output_file
```

Contenuto di configuration\_version\_output\_file.

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

Output:

```
{
  "ApplicationId": "339ohji",
  "ConfigurationProfileId": "ur8hx2f",
  "VersionNumber": "1",
  "ContentType": "application/json"
}
```

Per ulteriori informazioni, consulta [Informazioni sull'archivio di configurazione AWS AppConfig ospitato nella Guida per l'utente di AWS AppConfig](#).

- Per API i dettagli, vedere [CreateHostedConfigurationVersion](#) in AWS CLI Command Reference.

## delete-application

Il seguente esempio di codice mostra come utilizzare delete-application.

AWS CLI

Eliminazione di un'applicazione

L'delete-application esempio seguente elimina l'applicazione specificata.

```
aws appconfig delete-application \
--application-id 339ohji
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Fase 1: Creazione di un' AWS AppConfig applicazione](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [DeleteApplication](#) in AWS CLI Command Reference.

## delete-configuration-profile

Il seguente esempio di codice mostra come utilizzare `delete-configuration-profile`.

### AWS CLI

Per eliminare un profilo di configurazione

L'`delete-configuration-profile` esempio seguente elimina il profilo di configurazione specificato.

```
aws appconfig delete-configuration-profile \  
  --application-id 339ohji \  
  --configuration-profile-id ur8hx2f
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Fase 3: Creazione di una configurazione e di un profilo di configurazione](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [DeleteConfigurationProfile](#) in AWS CLI Command Reference.

## delete-deployment-strategy

Il seguente esempio di codice mostra come utilizzare `delete-deployment-strategy`.

### AWS CLI

Per eliminare una strategia di distribuzione

L'`delete-deployment-strategy` esempio seguente elimina la strategia di distribuzione specificata.

```
aws appconfig delete-deployment-strategy \  
  --deployment-strategy-id 1225qzk
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Fase 4: Creazione di una strategia di distribuzione](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [DeleteDeploymentStrategy](#) in AWS CLI Command Reference.

## delete-environment

Il seguente esempio di codice mostra come utilizzare `delete-environment`.

### AWS CLI

Per eliminare un ambiente

L'`delete-environment` esempio seguente elimina l'ambiente applicativo specificato.

```
aws appconfig delete-environment \  
  --application-id 339ohji \  
  --environment-id 54j1r29
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Fase 2: Creazione di un ambiente](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [DeleteEnvironment](#) in AWS CLI Command Reference.

## delete-extension-association

Il seguente esempio di codice mostra come utilizzare `delete-extension-association`.

### AWS CLI

Per eliminare un'associazione di estensioni

L'`delete-extension-association` esempio seguente elimina un'associazione di estensioni da AWS AppConfig.

```
aws appconfig delete-extension-association \  
  --region us-west-2 \  
  --extension-association-id a1b2c3d4
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consultate [Lavorare con AWS AppConfig le estensioni](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [DeleteExtensionAssociation](#) in AWS CLI Command Reference.

## delete-extension

Il seguente esempio di codice mostra come utilizzare `delete-extension`.

### AWS CLI

Per eliminare un'estensione

L'`delete-extension` esempio seguente elimina un'estensione da AWS AppConfig.

```
aws appconfig delete-extension \  
  --region us-west-2 \  
  --extension-identifier S3-backup-extension
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Lavorare con AWS AppConfig le estensioni](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [DeleteExtension](#) in AWS CLI Command Reference.

## delete-hosted-configuration-version

Il seguente esempio di codice mostra come utilizzare `delete-hosted-configuration-version`.

### AWS CLI

Per eliminare una versione di configurazione ospitata

L'`delete-hosted-configuration-version` esempio seguente elimina una versione di configurazione ospitata nell'archivio di configurazione AWS AppConfig ospitato.

```
aws appconfig delete-hosted-configuration-version \  
  --application-id 339ohji \  
  --configuration-profile-id ur8hx2f \  
  --version-number 1
```

Output:: Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Passaggio 3: Creazione di una configurazione e di un profilo di configurazione](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [DeleteHostedConfigurationVersion](#) in AWS CLI Command Reference.

## get-application

Il seguente esempio di codice mostra come utilizzare `get-application`.

### AWS CLI

Per elencare i dettagli di un'applicazione

L'`get-application` esempio seguente elenca i dettagli dell'applicazione specificata.

```
aws appconfig get-application \  
  --application-id 339ohji
```

Output:

```
{  
  "Description": "An application used for creating an example.",  
  "Id": "339ohji",  
  "Name": "example-application"  
}
```

Per ulteriori informazioni, vedere [How AWS AppConfig works](#) in the AWS AppConfig User Guide.

- Per API i dettagli, vedere [GetApplication](#) in AWS CLI Command Reference.

## get-configuration-profile

Il seguente esempio di codice mostra come utilizzare `get-configuration-profile`.

### AWS CLI

Per recuperare i dettagli del profilo di configurazione

L'`get-configuration-profile` esempio seguente restituisce i dettagli del profilo di configurazione specificato.

```
aws appconfig get-configuration-profile \  
  --application-id 339ohji \  
  --profile-id
```

```
--configuration-profile-id ur8hx2f
```

Output:

```
{
  "ApplicationId": "339ohji",
  "Id": "ur8hx2f",
  "Name": "Example-Configuration-Profile",
  "LocationUri": "ssm-parameter://Example-Parameter",
  "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role"
}
```

Per ulteriori informazioni, vedere [Fase 3: Creazione di una configurazione e di un profilo di configurazione](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [GetConfigurationProfile](#) in AWS CLI Command Reference.

## get-configuration

Il seguente esempio di codice mostra come utilizzare `get-configuration`.

### AWS CLI

Per recuperare i dettagli di configurazione

L'`get-configuration` esempio seguente restituisce i dettagli di configurazione dell'applicazione di esempio. Nelle chiamate successive a `get-configuration`, utilizzate il `client-configuration-version` parametro per aggiornare la configurazione dell'applicazione solo se la versione è cambiata. L'aggiornamento della configurazione solo quando la versione è stata modificata evita costi eccessivi dovuti alla chiamata a `get-configuration`.

```
aws appconfig get-configuration \
  --application "example-application" \
  --environment "Example-Environment" \
  --configuration "Example-Configuration-Profile" \
  --client-id "test-id" \
  configuration-output-file
```

Contenuto di `configuration-output-file`.

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

Output:

```
{
  "ConfigurationVersion": "1",
  "ContentType": "application/json"
}
```

Per ulteriori informazioni, vedere [Fase 6: Ricezione della configurazione nella Guida per l'utente.AWS AppConfig](#)

- Per API i dettagli, vedere [GetConfiguration](#) in AWS CLI Command Reference.

## get-deployment-strategy

Il seguente esempio di codice mostra come utilizzare `get-deployment-strategy`.

AWS CLI

Per recuperare i dettagli di una strategia di distribuzione

L'`get-deployment-strategy` seguente elenca i dettagli della strategia di distribuzione specificata.

```
aws appconfig get-deployment-strategy \
  --deployment-strategy-id 1225qzk
```

Output:

```
{
  "Id": "1225qzk",
  "Name": "Example-Deployment",
  "DeploymentDurationInMinutes": 15,
  "GrowthType": "LINEAR",
  "GrowthFactor": 25.0,
  "FinalBakeTimeInMinutes": 0,
  "ReplicateTo": "SSM_DOCUMENT"
}
```

Per ulteriori informazioni, vedere [Fase 4: Creazione di una strategia di distribuzione](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [GetDeploymentStrategy](#) in AWS CLI Command Reference.

## get-deployment

Il seguente esempio di codice mostra come utilizzare `get-deployment`.

### AWS CLI

Per recuperare i dettagli della distribuzione

L'`get-deployment` esempio seguente elenca i dettagli della distribuzione nell'applicazione nell'ambiente e nella distribuzione specificati.

```
aws appconfig get-deployment \  
  --application-id 339ohji \  
  --environment-id 54j1r29 \  
  --deployment-number 1
```

Output:

```
{  
  "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"
}

```

Per ulteriori informazioni, vedere [Fase 5: Distribuzione di una configurazione](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [GetDeployment](#) in AWS CLI Command Reference.

## get-environment

Il seguente esempio di codice mostra come utilizzare `get-environment`.

### AWS CLI

Per recuperare i dettagli dell'ambiente

L'`get-environment` esempio seguente restituisce i dettagli e lo stato dell'ambiente specificato.

```
aws appconfig get-environment \  
  --application-id 339ohji \  
  --environment-id 54j1r29
```

Output:

```
{  
  "ApplicationId": "339ohji",  
  "Id": "54j1r29",  
  "Name": "Example-Environment",  
  "State": "ReadyForDeployment"  
}
```

Per ulteriori informazioni, vedere [Fase 2: Creazione di un ambiente](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [GetEnvironment](#) in AWS CLI Command Reference.

## get-extension-association

Il seguente esempio di codice mostra come utilizzare `get-extension-association`.

### AWS CLI

Per ottenere i dettagli dell'associazione delle estensioni

L'`get-extension-association` esempio seguente visualizza informazioni su un'associazione di estensioni.

```
aws appconfig get-extension-association \  
  --region us-west-2 \  
  --extension-association-id a1b2c3d4
```

**Output:**

```
{
  "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
}
```

Per ulteriori informazioni, consultate [Lavorare con AWS AppConfig le estensioni](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [GetExtensionAssociation](#) in AWS CLI Command Reference.

**get-extension**

Il seguente esempio di codice mostra come utilizzare `get-extension`.

**AWS CLI**

Per ottenere i dettagli dell'estensione

L'`get-extension` esempio seguente visualizza informazioni su un'estensione.

```
aws appconfig get-extension \
  --region us-west-2 \
  --extension-identifier S3-backup-extension
```

**Output:**

```
{
  "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
    }
  }
}

```

Per ulteriori informazioni, consultate [Lavorare con AWS AppConfig le estensioni](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [GetExtension](#) in AWS CLI Command Reference.

## get-hosted-configuration-version

Il seguente esempio di codice mostra come utilizzare `get-hosted-configuration-version`.

### AWS CLI

Per recuperare i dettagli della configurazione ospitata

L'`get-hosted-configuration-version` esempio seguente recupera i dettagli di configurazione della configurazione ospitata. AWS AppConfig

```

aws appconfig get-hosted-configuration-version \
  --application-id 339ohji \
  --configuration-profile-id ur8hx2f \
  --version-number 1 \
  hosted-configuration-version-output

```

Contenuto di `hosted-configuration-version-output`.

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

Output:

```
{
  "ApplicationId": "339ohji",
  "ConfigurationProfileId": "ur8hx2f",
  "VersionNumber": "1",
  "ContentType": "application/json"
}
```

Per ulteriori informazioni, vedere [Informazioni sull'archivio di configurazione AWS AppConfig ospitato](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [GetHostedConfigurationVersion](#) in AWS CLI Command Reference.

## list-applications

Il seguente esempio di codice mostra come utilizzare `list-applications`.

### AWS CLI

Per elencare le applicazioni disponibili

L'`list-applications` seguente elenca le applicazioni disponibili nel tuo AWS account.

```
aws AppConfig list-applications
```

Output:

```
{
  "Items": [
    {
      "Id": "339ohji",
      "Name": "test-application",
      "Description": "An application used for creating an example."
    },
    {
      "Id": "rwalwu7",
      "Name": "Test-Application"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Fase 1: Creazione di un' AWS AppConfig applicazione](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [ListApplications](#) in AWS CLI Command Reference.

## list-configuration-profiles

Il seguente esempio di codice mostra come utilizzare `list-configuration-profiles`.

### AWS CLI

Per elencare i profili di configurazione disponibili

L'`list-configuration-profiles` seguente elenca i profili di configurazione disponibili per l'applicazione specificata.

```
aws appconfig list-configuration-profiles \  
  --application-id 339ohji
```

Output:

```
{  
  "Items": [  
    {  
      "ApplicationId": "339ohji",  
      "Id": "ur8hx2f",  
      "Name": "Example-Configuration-Profile",  
      "LocationUri": "ssm-parameter://Example-Parameter"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Fase 3: Creazione di una configurazione e di un profilo di configurazione](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [ListConfigurationProfiles](#) in AWS CLI Command Reference.

## list-deployment-strategies

Il seguente esempio di codice mostra come utilizzare `list-deployment-strategies`.

### AWS CLI

Per elencare le strategie di distribuzione disponibili

L'`list-deployment-strategies` seguente elenca le strategie di distribuzione disponibili nel tuo AWS account.

```
aws appconfig list-deployment-strategies
```

Output:

```
{
  "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"
    }
]
}
```

Per ulteriori informazioni, consulta [Fase 4: Creazione di una strategia di distribuzione](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [ListDeploymentStrategies](#) in AWS CLI Command Reference.

## list-deployments

Il seguente esempio di codice mostra come utilizzare `list-deployments`.

### AWS CLI

Per elencare le distribuzioni disponibili

L'`list-deployments` esempio seguente elenca le distribuzioni disponibili nell'AWS account per l'applicazione e l'ambiente specificati.

```
aws appconfig list-deployments \
  --application-id 339ohji \
  --environment-id 54j1r29
```

Output:

```
{
  "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"
    }
]
}
```

Per ulteriori informazioni, vedere [Fase 5: Distribuzione di una configurazione](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [ListDeployments](#) in AWS CLI Command Reference.

## list-environments

Il seguente esempio di codice mostra come utilizzare `list-environments`.

### AWS CLI

Per elencare gli ambienti disponibili

L'`list-environments` esempio seguente elenca gli ambienti disponibili nell' AWS account per l'applicazione specificata.

```
aws appconfig list-environments \
  --application-id 339ohji
```

Output:

```
{
  "Items": [
    {
      "ApplicationId": "339ohji",
      "Id": "54j1r29",
      "Name": "Example-Environment",
      "State": "ReadyForDeployment"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Fase 2: Creazione di un ambiente](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [ListEnvironments](#) in AWS CLI Command Reference.

## list-extension-associations

Il seguente esempio di codice mostra come utilizzare `list-extension-associations`.

### AWS CLI

Per elencare tutte le associazioni di AWS AppConfig estensioni presenti nel tuo AWS account per una AWS regione

L'`list-extension-associations` esempio seguente elenca tutte le associazioni di AWS AppConfig estensione per l' AWS account corrente in una AWS regione specifica.

```
aws appconfig list-extension-associations \  
  --region us-west-2
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Lavorare con AWS AppConfig le estensioni](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [ListExtensionAssociations](#) in AWS CLI Command Reference.

## list-extensions

Il seguente esempio di codice mostra come utilizzare `list-extensions`.

### AWS CLI

Per elencare tutte AWS AppConfig le estensioni presenti nel tuo AWS account per una AWS regione

L'list-extensionsesempio seguente elenca tutte AWS AppConfig le estensioni per l' AWS account corrente in una AWS regione specifica. Il comando restituisce estensioni personalizzate e AWS create.

```
aws appconfig list-extensions \  
  --region us-west-2
```

Output:

```
{  
  "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."
    }
]
}

```

Per ulteriori informazioni, consulta [Lavorare con AWS AppConfig le estensioni](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [ListExtensions](#) in AWS CLI Command Reference.

## list-hosted-configuration-versions

Il seguente esempio di codice mostra come utilizzare `list-hosted-configuration-versions`.

### AWS CLI

Per elencare le versioni di configurazione ospitate disponibili

L'`list-hosted-configuration-versions` seguente elenca le versioni delle configurazioni ospitate nell'archivio di configurazione AWS AppConfig ospitato per l'applicazione e il profilo di configurazione specificati.

```
aws appconfig list-hosted-configuration-versions \  
  --application-id 339ohji \  
  --configuration-profile-id ur8hx2f
```

Output:

```
{  
  "Items": [  
    {  
      "ApplicationId": "339ohji",  
      "ConfigurationProfileId": "ur8hx2f",  
      "VersionNumber": 1,  
      "ContentType": "application/json"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Informazioni sull'archivio di configurazione AWS AppConfig ospitato](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [ListHostedConfigurationVersions](#) in AWS CLI Command Reference.

## **list-tags-for-resource**

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

AWS CLI

Per elencare i tag di un'applicazione

L'`list-tags-for-resource` seguente elenca i tag di un'applicazione specificata.

```
aws appconfig list-tags-for-resource \  
  --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji
```

Output:

```
{
  "Tags": {
    "group1": "1"
  }
}
```

Per ulteriori informazioni, vedere [Fase 1: Creazione di un' AWS AppConfig applicazione](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## start-deployment

Il seguente esempio di codice mostra come utilizzare `start-deployment`.

### AWS CLI

Per avviare una distribuzione della configurazione

L'`start-deployment` seguente avvia una distribuzione nell'applicazione utilizzando l'ambiente, la strategia di distribuzione e il profilo di configurazione specificati.

```
aws appconfig start-deployment \
  --application-id 339ohji \
  --environment-id 54j1r29 \
  --deployment-strategy-id 1225qzk \
  --configuration-profile-id ur8hx2f \
  --configuration-version 1
```

Output:

```
{
  "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"
  }

```

Per ulteriori informazioni, vedere [Fase 5: Distribuzione di una configurazione](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [StartDeployment](#) in AWS CLI Command Reference.

## stop-deployment

Il seguente esempio di codice mostra come utilizzare `stop-deployment`.

### AWS CLI

Per interrompere la distribuzione della configurazione

L'`stop-deployment` seguente interrompe la distribuzione di una configurazione dell'applicazione nell'ambiente specificato.

```

aws appconfig stop-deployment \
  --application-id 339ohji \
  --environment-id 54j1r29 \
  --deployment-number 2

```

Output:

```

{
  "DeploymentNumber": 0,
  "DeploymentDurationInMinutes": 0,
  "GrowthFactor": 0.0,

```

```
"FinalBakeTimeInMinutes": 0,  
"PercentageComplete": 0.0  
}
```

Per ulteriori informazioni, vedere [Fase 5: Distribuzione di una configurazione](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [StopDeployment](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare un'applicazione

L'`tag-resource` esempio seguente contrassegna una risorsa dell'applicazione.

```
aws appconfig tag-resource \  
  --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji \  
  --tags '{"group1" : "1"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Fase 1: Creazione di un' AWS AppConfig applicazione](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da un'applicazione

L'`untag-resource` esempio seguente rimuove il tag `group1` dall'applicazione specificata.

```
aws appconfig untag-resource \  
  --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji \  
  --tag-key group1
```



```
--resource-arn arn:aws:appconfig:us-east-1:111122223333:application/339ohji \  
--tag-keys '["group1"]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Fase 1: Creazione di un' AWS AppConfig applicazione](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-application

Il seguente esempio di codice mostra come utilizzare `update-application`.

### AWS CLI

Per aggiornare un'applicazione

L'`update-application` esempio seguente aggiorna il nome dell'applicazione specificata.

```
aws appconfig update-application \  
--application-id 339ohji \  
--name "Example-Application"
```

Output:

```
{  
  "Id": "339ohji",  
  "Name": "Example-Application",  
  "Description": "An application used for creating an example."  
}
```

Per ulteriori informazioni, vedere [Fase 1: Creazione di un' AWS AppConfig applicazione](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [UpdateApplication](#) in AWS CLI Command Reference.

## update-configuration-profile

Il seguente esempio di codice mostra come utilizzare `update-configuration-profile`.



```
--final-bake-time-in-minutes 20
```

Output:

```
{
  "Id": "1225qzk",
  "Name": "Example-Deployment",
  "DeploymentDurationInMinutes": 15,
  "GrowthType": "LINEAR",
  "GrowthFactor": 25.0,
  "FinalBakeTimeInMinutes": 20,
  "ReplicateTo": "SSM_DOCUMENT"
}
```

Per ulteriori informazioni, consulta [Fase 4: Creazione di una strategia di distribuzione](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [UpdateDeploymentStrategy](#) in AWS CLI Command Reference.

## update-environment

Il seguente esempio di codice mostra come utilizzare `update-environment`.

### AWS CLI

Per aggiornare un ambiente

L'`update-environment` esempio seguente aggiorna la descrizione di un ambiente.

```
aws appconfig update-environment \
  --application-id 339ohji \
  --environment-id 54j1r29 \
  --description "An environment for examples."
```

Output:

```
{
  "ApplicationId": "339ohji",
  "Id": "54j1r29",
  "Name": "Example-Environment",
  "Description": "An environment for examples.",
}
```

```
"State": "RolledBack"
}
```

Per ulteriori informazioni, vedere [Fase 2: Creazione di un ambiente](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [UpdateEnvironment](#) in AWS CLI Command Reference.

## update-extension-association

Il seguente esempio di codice mostra come utilizzare `update-extension-association`.

### AWS CLI

Per aggiornare un'associazione di AWS AppConfig estensioni

L'`update-extension-association` esempio seguente aggiunge un nuovo valore di parametro a un'associazione di estensione in AWS AppConfig.

```
aws appconfig update-extension-association \
  --region us-west-2 \
  --extension-association-id a1b2c3d4 \
  --parameters S3bucket=FinanceMobileApp
```

Output:

```
{
  "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
}
```

Per ulteriori informazioni, vedete [Lavorare con AWS AppConfig le estensioni](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [UpdateExtensionAssociation](#) in AWS CLI Command Reference.

## update-extension

Il seguente esempio di codice mostra come utilizzare `update-extension`.

### AWS CLI

Per aggiornare un' AWS AppConfig estensione

L'`update-extension` esempio seguente aggiunge un parametro aggiuntivo `Key` a un'estensione in AWS AppConfig.

```
aws appconfig update-extension \  
  --region us-west-2 \  
  --extension-identifier S3-backup-extension \  
  --parameters S3bucket={Required=true}, CampaignID={Required=false}
```

Output:

```
{  
  "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  
    }  
  }  
}
```

Per ulteriori informazioni, consultate [Lavorare con AWS AppConfig le estensioni](#) nella Guida AWS AppConfig per l'utente.

- Per API i dettagli, vedere [UpdateExtension](#) in AWS CLI Command Reference.

## **validate-configuration**

Il seguente esempio di codice mostra come utilizzare `validate-configuration`.

### AWS CLI

Per convalidare una configurazione

L'`validate-configuration` esempio seguente utilizza i validatori in un profilo di configurazione per convalidare una configurazione.

```
aws appconfig validate-configuration \  
  --application-id abc1234 \  
  --configuration-profile-id ur8hx2f \  
  --configuration-version 1
```

Il comando non produce alcun output.

Per ulteriori informazioni, vedere [Fase 3: Creazione di una configurazione e di un profilo di configurazione](#) nella Guida per l'AWS AppConfig utente.

- Per API i dettagli, vedere [ValidateConfiguration](#) in AWS CLI Command Reference.

## Esempi di Application Auto Scaling utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Application Auto Scaling.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto degli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **delete-scaling-policy**

Il seguente esempio di codice mostra come utilizzare `delete-scaling-policy`.

#### AWS CLI

Per eliminare una politica di scalabilità

Questo esempio elimina una politica di scalabilità per l'app web del ECS servizio Amazon in esecuzione nel cluster predefinito.

Comando:

```
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
```

- Per i API dettagli, consulta [DeleteScalingPolicy](#) Command Reference.AWS CLI

### **delete-scheduled-action**

Il seguente esempio di codice mostra come utilizzare `delete-scheduled-action`.

#### AWS CLI

Per eliminare un'operazione pianificata

L'`delete-scheduled-action` esempio seguente elimina l'azione pianificata specificata dalla flotta Amazon AppStream 2.0 specificata:

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Dimensionamento pianificato](#) nella Guida per l'utente di Application Auto Scaling.

- Per API i dettagli, consulta AWS CLI Command [DeleteScheduledAction](#) Reference.

## deregister-scalable-target

Il seguente esempio di codice mostra come utilizzare `deregister-scalable-target`.

### AWS CLI

Per annullare la registrazione di un target scalabile

Questo esempio annulla la registrazione di una destinazione scalabile per un ECS servizio Amazon chiamato `web-app` in esecuzione nel cluster predefinito.

Comando:

```
aws application-autoscaling deregister-scalable-target --service-namespace ecs --scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-app
```

Questo esempio annulla la registrazione di una destinazione scalabile per una risorsa personalizzata. Il `custom-resource-id file.txt` contiene una stringa che identifica l'ID risorsa, che, per una risorsa personalizzata, è il percorso verso la risorsa personalizzata attraverso l'endpoint Amazon API Gateway.

Comando:

```
aws application-autoscaling deregister-scalable-target --service-namespace custom-resource --scalable-dimension custom-resource:ResourceType:Property --resource-id file://~/custom-resource-id.txt
```

Contenuto del file.txt: `custom-resource-id`

```
https://example.execute-api.us-west-2.amazonaws.com/prod/scalableTargetDimensions/1-23456789
```

- Per API i dettagli, vedere [DeregisterScalableTarget](#) in AWS CLI Command Reference.



## describe-scalable-targets

Il seguente esempio di codice mostra come utilizzare `describe-scalable-targets`.

### AWS CLI

Per descrivere obiettivi scalabili

L'esempio seguente descrive le destinazioni scalabili per il namespace del ecs servizio.

```
aws application-autoscaling describe-scalable-targets \
  --service-namespace ecs
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [AWS i servizi che è possibile utilizzare con Application Auto Scaling nella Application Auto Scaling User Guide](#).

- Per API i dettagli, vedere [DescribeScalableTargets](#) in AWS CLI Command Reference.

## describe-scaling-activities

Il seguente esempio di codice mostra come utilizzare `describe-scaling-activities`.

### AWS CLI

Esempio 1: per descrivere le attività di scalabilità per il servizio Amazon ECS specificato

L'esempio seguente descrive le attività di scalabilità per un ECS servizio Amazon chiamato `web-app` che è in esecuzione nel `default` cluster. L'output mostra un'attività di scalabilità avviata da una politica di scalabilità.

```
aws application-autoscaling describe-scaling-activities \
  --service-namespace ecs \
  --resource-id service/default/web-app
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Scaling activities for Application Auto Scaling nella Application Auto Scaling User Guide](#).

Esempio 2: Per descrivere le attività di ridimensionamento per la tabella DynamoDB specificata

L'output di `aws application-autoscaling describe-scaling-activities` seguente descrive le attività di ridimensionamento per una tabella DynamoDB chiamata `TestTable`. L'output mostra le attività di ridimensionamento avviate da due diverse azioni pianificate.

```
aws application-autoscaling describe-scaling-activities \
  --service-namespace dynamodb \
  --resource-id table/TestTable
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, vedere [Scaling activities for Application Auto Scaling nella Application Auto Scaling User Guide](#).

- Per API i dettagli, vedere [DescribeScalingActivities](#) in AWS CLI Command Reference.

## describe-scaling-policies

Il seguente esempio di codice mostra come utilizzare `describe-scaling-policies`.

### AWS CLI

Per descrivere le politiche di scalabilità

Questo comando di esempio descrive le politiche di scalabilità per lo spazio dei nomi del servizio `ecs`.

Comando:

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs
```

## Output:

```
{
  "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"
}
]
}

```

- Per i API dettagli, vedere [DescribeScalingPolicies](#) in Command Reference.AWS CLI

## describe-scheduled-actions

Il seguente esempio di codice mostra come utilizzare `describe-scheduled-actions`.

### AWS CLI

Per descrivere le azioni pianificate

L'`describe-scheduled-actions` esempio seguente visualizza i dettagli delle azioni pianificate per il namespace del servizio specificato:

```
aws application-autoscaling describe-scheduled-actions \
  --service-namespace dynamodb
```

Output:

```
{
  "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"
}
]
}

```

Per ulteriori informazioni, consulta [Dimensionamento pianificato](#) nella Guida per l'utente di Application Auto Scaling.

- Per API i dettagli, vedere [DescribeScheduledActions](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag per un target scalabile

L'`list-tags-for-resource` seguente elenca i nomi e i valori delle chiavi dei tag associati alla destinazione scalabile specificata da its. ARN

```
aws application-autoscaling list-tags-for-resource \  
  --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-  
target/1234abcd56ab78cd901ef1234567890ab123
```

Output:

```
{  
  "Tags": {  
    "environment": "production"  
  }  
}
```

Per ulteriori informazioni, consulta il [supporto per l'etichettatura per Application Auto Scaling nella Application Auto Scaling User Guide](#).

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## put-scaling-policy

Il seguente esempio di codice mostra come utilizzare `put-scaling-policy`.

### AWS CLI

Esempio 1: applicazione di una policy di dimensionamento con monitoraggio degli obiettivi con uno specifico parametro di default

L'`put-scaling-policy` seguente applica una politica di scalabilità del tracciamento degli obiettivi con una specifica metrica predefinita a un ECS servizio Amazon chiamato `web-app` nel cluster predefinito. La policy mantiene l'CPU utilizzo medio del servizio al 75%, con periodi di recupero della scalabilità orizzontale e intermedia di 60 secondi. L'output contiene i nomi ARNs e i nomi dei due CloudWatch allarmi creati per tuo conto.

```
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
```



Questo esempio presuppone che nella directory corrente sia presente un file `config.json` con il seguente contenuto:

```
{
  "TargetValue": 75.0,
  "PredefinedMetricSpecification": {
    "PredefinedMetricType": "ECSServiceAverageCPUUtilization"
  },
  "ScaleOutCooldown": 60,
  "ScaleInCooldown": 60
}
```

Output:

```
{
  "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"
    }
  ]
}
```

**Esempio 2: applicazione di una policy di dimensionamento con monitoraggio degli obiettivi con uno specifico parametro personalizzato**

L'`put-scaling-policy` esempio seguente applica una politica di scalabilità del tracciamento degli obiettivi con una specifica metrica personalizzata a un ECS servizio Amazon chiamato `web-app` nel cluster predefinito. La policy mantiene l'utilizzo medio del servizio al 75%, con periodi di

recupero della scalabilità orizzontale e intermedia di 60 secondi. L'output contiene i nomi ARNs e i nomi dei due CloudWatch allarmi creati per tuo conto.

```
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
```

Questo esempio presuppone che nella directory corrente sia presente un file config.json con il seguente contenuto:

```
{  
  "TargetValue":75.0,  
  "CustomizedMetricSpecification":{  
    "MetricName":"MyUtilizationMetric",  
    "Namespace":"MyNamespace",  
    "Dimensions": [  
      {  
        "Name":"MyOptionalMetricDimensionName",  
        "Value":"MyOptionalMetricDimensionValue"  
      }  
    ],  
    "Statistic":"Average",  
    "Unit":"Percent"  
  },  
  "ScaleOutCooldown": 60,  
  "ScaleInCooldown": 60  
}
```

Output:

```
{  
  "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"
  }
]
}

```

Esempio 3: applicazione di una policy di dimensionamento con monitoraggio degli obiettivi per il solo aumento orizzontale

L'`put-scaling-policy` seguente applica una politica di dimensionamento del tracciamento degli obiettivi a un ECS servizio Amazon chiamato `web-app` nel cluster predefinito. La policy viene utilizzata per scalare orizzontalmente il ECS servizio quando la `RequestCountPerTarget` metrica dell'Application Load Balancer supera la soglia. L'output contiene il nome ARN e il nome dell' CloudWatch allarme creato per tuo conto.

```

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

```

Contenuto di `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
}

```

```
}

```

### Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Target Tracking Scaling Policies for Application Auto Scaling](#) nella AWS Application Auto Scaling User Guide.

- Per API i dettagli, consulta Command [PutScalingPolicy](#)Reference AWS CLI .

## put-scheduled-action

Il seguente esempio di codice mostra come utilizzare `put-scheduled-action`.

### AWS CLI

Per aggiungere un'azione pianificata a una tabella DynamoDB

Questo esempio aggiunge un'azione pianificata a una tabella DynamoDB `TestTable` chiamata per scalare out in base a una pianificazione ricorrente. Nella pianificazione specificata (ogni giorno alle 12:15UTC), se la capacità corrente è inferiore al valore specificato per `MinCapacity`, Application Auto Scaling viene ridimensionato fino al valore specificato da `MinCapacity`

Comando:

```
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
```

Per ulteriori informazioni, vedere Scheduled Scaling nella Application Auto Scaling User Guide.

- Per API i dettagli, vedere [PutScheduledAction](#) in AWS CLI Command Reference.

## register-scalable-target

Il seguente esempio di codice mostra come utilizzare `register-scalable-target`.

### AWS CLI

Esempio 1: registrare un ECS servizio come destinazione scalabile

L'`register-scalable-target` seguente registra un ECS servizio Amazon con Application Auto Scaling. Aggiunge inoltre un tag con il nome della chiave `environment` e il valore `production` alla destinazione scalabile.

```
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
```

Output:

```
{
  "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Per esempi di altri AWS servizi e risorse personalizzate, consultate gli argomenti relativi [AWS ai servizi che è possibile utilizzare con Application Auto Scaling nella Application Auto Scaling User Guide](#).

Esempio 2: sospendere le attività di scalabilità per un target scalabile

L'`register-scalable-target` seguente sospende le attività di scalabilità per un target scalabile esistente.

```
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
```

Output:

```
{
  "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Per ulteriori informazioni, vedere [Suspending and resuming scaling for Application Auto Scaling nella Application Auto Scaling User Guide](#).

Esempio 3: riprendere le attività di scalabilità per un target scalabile

L'`register-scalable-target` seguente riprende le attività di scalabilità per un target scalabile esistente.

```
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
```

Output:

```
{
  "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

Per ulteriori informazioni, vedere [Suspending and resuming scaling for Application Auto Scaling nella Application Auto Scaling User Guide](#).

- Per i API dettagli, vedere in Command Reference. [RegisterScalableTarget](#) AWS CLI

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere un tag a un target scalabile

L'`tag-resource` esempio seguente aggiunge un tag con il nome della chiave `environment` e il valore `production` alla destinazione scalabile specificata da `its`. ARN

```
aws application-autoscaling tag-resource \  
  --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-  
target/1234abcd56ab78cd901ef1234567890ab123 \  
  --tags environment=production
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta il [supporto per l'etichettatura per Application Auto Scaling nella Application Auto Scaling User Guide](#).

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da un target scalabile

L'`untag-resource` esempio seguente rimuove la coppia di tag con il nome della chiave `environment` dalla destinazione scalabile specificata da `its`. ARN

```
aws application-autoscaling untag-resource \  
  --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-  
target/1234abcd56ab78cd901ef1234567890ab123 \  
  --tag-keys "environment"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta il [supporto per l'etichettatura per Application Auto Scaling nella Application Auto Scaling User Guide](#).

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## Esempi di Application Discovery Service utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il servizio AWS Command Line Interface with Application Discovery Service.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni illustrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto degli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **describe-agents**

Il seguente esempio di codice mostra come utilizzare `describe-agents`.

#### AWS CLI

Descrivi gli agenti con `collectionStatus` stati specifici

Questo comando di esempio descrive gli agenti di raccolta con lo stato di raccolta `STARTED` o `STOPPED`.

Comando:

```
aws discovery describe-agents --filters
  name="collectionStatus",values="STARTED","STOPPED",condition="EQUALS" --max-
  results 3
```

Output:

```
{
  "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"
  }
]
}

```

- Per API i dettagli, vedere [DescribeAgents](#) in AWS CLI Command Reference.

## describe-configurations

Il seguente esempio di codice mostra come utilizzare `describe-configurations`.

### AWS CLI

Descrivi le configurazioni degli asset selezionati

Questo comando di esempio descrive le configurazioni di due server specificati. L'azione rileva il tipo di risorsa dall'ID di configurazione. È consentito un solo tipo di risorsa per comando.

Comando:

```
aws discovery describe-configurations --configuration-ids "d-  
server-099385097ef9fbcfb" "d-server-0c4f2dd1fee22c6c1"
```

Output:

```
{
  "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": "1547210.1333333333",
"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": "14.733333333333338",
"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"
    }
]
}

```

Descrivi le configurazioni degli asset selezionati

Questo comando di esempio descrive le configurazioni di due applicazioni specificate. L'azione rileva il tipo di risorsa dall'ID di configurazione. È consentito un solo tipo di risorsa per comando.

Comando:

```

aws discovery describe-configurations --configuration-ids "d-
application-0ac39bc0e4fad0e42" "d-application-02444a45288013764q"

```

Output:

```

{
  "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"
    }
]
}

```

- Per API i dettagli, consultate [DescribeConfigurations AWS CLI Command Reference](#).

## list-configurations

Il seguente esempio di codice mostra come utilizzare `list-configurations`.

### AWS CLI

Per elencare tutti i server rilevati che soddisfano una serie di condizioni di filtro

Questo comando di esempio elenca i server rilevati che corrispondono a uno dei due modelli di hostname e che non eseguono Ubuntu.

Comando:

```

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"

```

Output:

```

{
  "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"
  }
]
}
```

- Per API i dettagli, vedere [ListConfigurations](#) in AWS CLI Command Reference.

## AppRegistry esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AppRegistry.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **associate-attribute-group**

Il seguente esempio di codice mostra come utilizzare `associate-attribute-group`.

AWS CLI

Per associare un gruppo di attributi

L'esempio seguente associa un gruppo di attributi specifico del tuo AWS account a un'applicazione specifica del tuo AWS account.

```
aws servicecatalog-appregistry associate-attribute-group \  
  --application "ExampleApplication" \  
  --attribute-group "ExampleAttributeGroup"
```

Output:

```
{  
  "applicationArn": "arn:aws:servicecatalog:us-west-2:813737243517:/  
applications/0ars38r6btoohvpvd9gqrptt91",  
  "attributeGroupArn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-  
groups/01sj5xdwhbw54kejwnt09fnpcl"  
}
```

Per ulteriori informazioni, vedere [Associating and Dissociating Attributing Groups](#) nella AWS Service Catalog Administrator Guide. AppRegistry

- Per API i dettagli, vedere [AssociateAttributeGroup](#) in AWS CLI Command Reference.

## create-application

Il seguente esempio di codice mostra come utilizzare `create-application`.

AWS CLI

Per creare un'applicazione

L'`create-application` esempio seguente crea una nuova applicazione nel tuo AWS account.

```
aws servicecatalog-appregistry create-application \  
  --name "ExampleApplication"
```

Output:

```
{  
  "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": {}
  }
}

```

Per ulteriori informazioni, vedere [Creazione di applicazioni](#) nella AWS Service Catalog AppRegistry Administrator Guide.

- Per API i dettagli, vedere [CreateApplication](#) in AWS CLI Command Reference.

## create-attribute-group

Il seguente esempio di codice mostra come utilizzare `create-attribute-group`.

### AWS CLI

Per creare un gruppo di attributi

L'esempio seguente crea un nuovo gruppo di attributi nel tuo AWS account.

```

aws servicecatalog-appregistry create-attribute-group \
  --name "ExampleAttributeGroup" \
  --attributes '{"SomeKey1":"SomeValue1","SomeKey2":"SomeValue2"}'

```

Output:

```

{
  "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": {}
  }
}

```

Per ulteriori informazioni, vedere [Creazione di gruppi di attributi](#) nella AWS Service Catalog AppRegistry Administrator Guide.



- Per API i dettagli, vedere [CreateAttributeGroup](#) in AWS CLI Command Reference.

## delete-application

Il seguente esempio di codice mostra come utilizzare `delete-application`.

### AWS CLI

Eliminazione di un'applicazione

L'`delete-application` esempio seguente elimina un'applicazione specifica dall' AWS account.

```
aws servicecatalog-appregistry delete-application \  
  --application "ExampleApplication3"
```

Output:

```
{  
  "application": {  
    "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Eliminazione di applicazioni](#) nella AWS Service Catalog AppRegistry Administrator Guide.

- Per API i dettagli, vedere [DeleteApplication](#) in AWS CLI Command Reference.

## delete-attribute-group

Il seguente esempio di codice mostra come utilizzare `delete-attribute-group`.

### AWS CLI

Esempio 8: Per eliminare un gruppo di attributi

L'`delete-attribute-group` seguente elimina un gruppo di attributi specifico nel tuo AWS account.

```
aws servicecatalog-appregistry delete-attribute-group \  
--attribute-group "ExampleAttributeGroup3"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Eliminazione dei gruppi di attributi](#) nella AWS Service Catalog AppRegistry Administrator Guide.

- Per API i dettagli, vedere [DeleteAttributeGroup](#) in AWS CLI Command Reference.

## get-application

Il seguente esempio di codice mostra come utilizzare `get-application`.

AWS CLI

Per ottenere un'applicazione

L'`get-application` seguente recupera le informazioni sui metadati relative a un'applicazione specifica nel tuo AWS account.

```
aws servicecatalog-appregistry get-application \  
--application "ExampleApplication"
```

Output:

```
{
```

```

    "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"
      }
    }
  }
}

```

Per ulteriori informazioni, vedere [Using Application details](#) nella AWS Service Catalog AppRegistry Administrator Guide.

- Per API i dettagli, vedere [GetApplication](#) in AWS CLI Command Reference.

## get-attribute-group

Il seguente esempio di codice mostra come utilizzare `get-attribute-group`.

### AWS CLI

Per ottenere un gruppo di attributi

L'`get-attribute-group` esempio seguente recupera un gruppo di attributi specifico nel tuo AWS account.

```

aws servicecatalog-appregistry get-attribute-group \
  --attribute-group ExampleAttributeGroup

```

Output:

```

{
  "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"
}
}
```

Per ulteriori informazioni, vedere [Gestione dei metadati per i gruppi di attributi](#) nella AWS Service Catalog AppRegistry Administrator Guide.

- Per API i dettagli, vedere [GetAttributeGroup](#) in AWS CLI Command Reference.

## list-applications

Il seguente esempio di codice mostra come utilizzare `list-applications`.

### AWS CLI

Per elencare le applicazioni

L'`list-application`sempio seguente recupera un elenco di tutte le applicazioni del tuo AWS account.

```
aws servicecatalog-appregistry list-applications
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, vedere [Visualizzazione dei dettagli dell'applicazione](#) nella AWS Service Catalog AppRegistry Administrator Guide.

- Per API i dettagli, vedere [ListApplications](#) in AWS CLI Command Reference.

## list-associated-attribute-groups

Il seguente esempio di codice mostra come utilizzare `list-associated-attribute-groups`.

### AWS CLI

Per elencare i gruppi di attributi associati

L'`list-associated-attribute-groups` esempio seguente recupera un elenco di tutti i gruppi di attributi AWS dell'account associati a un'applicazione specifica dell' AWS account.

```
aws servicecatalog-appregistry list-associated-attribute-groups \
  --application "ExampleApplication"
```

Output:

```
{
  "attributeGroups": [
```

```
    "01sj5xdwhbw54kejwnt09fnpc1"  
  ]  
}
```

Per ulteriori informazioni, vedere [Associating and Dissociating Attributing Groups](#) nella AWS Service Catalog Administrator Guide. AppRegistry

- Per API i dettagli, vedere [ListAssociatedAttributeGroups](#) in AWS CLI Command Reference.

## list-attribute-groups-for-application

Il seguente esempio di codice mostra come utilizzare `list-attribute-groups-for-application`.

### AWS CLI

Per elencare i gruppi di attributi per un'applicazione

L'`list-attribute-groups-for-application` esempio seguente elenca i dettagli di tutti i gruppi di attributi AWS dell'account associati a un'applicazione specifica dell' AWS account.

```
aws servicecatalog-appregistry list-attribute-groups-for-application \  
  --application "ExampleApplication"
```

Output:

```
{  
  "attributeGroupsDetails": [  
    {  
      "id": "01sj5xdwhbw54kejwnt09fnpc1",  
      "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-  
groups/01sj5xdwhbw54kejwnt09fnpc1",  
      "name": "ExampleAttributeGroup"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Visualizzazione dei dettagli del gruppo di attributi](#) nella AWS Service Catalog AppRegistry Administrator Guide.

- Per API i dettagli, vedere [ListAttributeGroupsForApplication](#) in AWS CLI Command Reference.

## list-attribute-groups

Il seguente esempio di codice mostra come utilizzare `list-attribute-groups`.

### AWS CLI

Per elencare i gruppi di attributi

L'`list-attribute-groups` seguente recupera un elenco di tutti i gruppi di attributi presenti nell' AWS account.

```
aws servicecatalog-appregistry list-attribute-groups
```

Output:

```
{
  "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"
    }
  ]
}
```

```
}
```

Per ulteriori informazioni, vedere [Visualizzazione dei dettagli del gruppo di attributi](#) nella AWS Service Catalog AppRegistry Administrator Guide.

- Per API i dettagli, vedere [ListAttributeGroups](#) in AWS CLI Command Reference.

## update-application

Il seguente esempio di codice mostra come utilizzare `update-application`.

### AWS CLI

Per aggiornare un'applicazione

L'`update-application` seguente aggiorna un'applicazione specifica nell' AWS account per includere una descrizione.

```
aws servicecatalog-appregistry update-application \  
  --application "ExampleApplication" \  
  --description "This is an example application"
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, vedere [Modifica delle applicazioni](#) nella AWS Service Catalog AppRegistry Administrator Guide.

- Per API i dettagli, vedere [UpdateApplication](#) in AWS CLI Command Reference.



## update-attribute-group

Il seguente esempio di codice mostra come utilizzare `update-attribute-group`.

### AWS CLI

Per aggiornare un gruppo di attributi

L'`update-attribute-group` seguente aggiorna un gruppo di attributi specifico nel tuo AWS account per includere una descrizione.

```
aws servicecatalog-appregistry update-attribute-group \
  --attribute-group "ExampleAttributeGroup" \
  --description "This is an example attribute group"
```

Output:

```
{
  "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"
    }
  }
}
```

Per ulteriori informazioni, vedere [Modifica dei gruppi di attributi](#) nella AWS Service Catalog AppRegistry Administrator Guide.

- Per API i dettagli, vedere [UpdateAttributeGroup](#) in AWS CLI Command Reference.

## Esempi di Athena che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Athena.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **batch-get-named-query**

Il seguente esempio di codice mostra come utilizzare `batch-get-named-query`.

AWS CLI

Per restituire informazioni su più di una query

L'`batch-get-named-query` esempio seguente restituisce informazioni sulle interrogazioni denominate che hanno il valore specificato 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
```

Output:

```
{
  "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": []
}

```

Per ulteriori informazioni, consulta [Running SQL Queries Using Amazon Athena nella Amazon Athena User Guide](#).

- Per API i dettagli, consulta Command [BatchGetNamedQuery](#)Reference AWS CLI .

## batch-get-query-execution

Il seguente esempio di codice mostra come utilizzare `batch-get-query-execution`.

### AWS CLI

Per restituire informazioni su una o più esecuzioni di query

L'esempio seguente restituisce le informazioni sull'esecuzione delle query per le quali è stata specificata la query. IDs

```

aws athena batch-get-query-execution \
  --query-execution-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222

```

## Output:

```
{
  "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": []
}

```

Per ulteriori informazioni, consulta [Running SQL Queries Using Amazon Athena nella Amazon Athena User Guide](#).

- Per API i dettagli, consulta Command [BatchGetQueryExecution](#)Reference AWS CLI .

## create-data-catalog

Il seguente esempio di codice mostra come utilizzare `create-data-catalog`.

### AWS CLI

Per creare un catalogo di dati

L'`create-data-catalog`esempio seguente crea il catalogo di `dynamo_db_catalog` dati.

```

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

```

Questo comando non produce alcun output. Per vedere il risultato, usa `aws athena get-data-catalog --name dynamo_db_catalog`.

Per ulteriori informazioni, consulta [Registrazione di un catalogo: create-data-catalog nella Guida per l'utente di Amazon Athena](#).

- Per API i dettagli, consulta AWS CLI Command [CreateDataCatalogReference](#).

## create-named-query

Il seguente esempio di codice mostra come utilizzare `create-named-query`.

### AWS CLI

Per creare una query denominata

L'esempio seguente crea una query salvata nel `AthenaAdmin` gruppo di lavoro che interroga la `flights_parquet` tabella per i voli da Seattle a New York nel gennaio 2016, la cui partenza e arrivo sono stati entrambi ritardati di oltre dieci minuti. Poiché i valori dei codici aeroportuali nella tabella sono stringhe che includono virgolette doppie (ad esempio, "SEA«), vengono sostituiti da barre rovesciate e racchiusi tra virgolette singole.

```
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-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
```

Output:

```
{  
  "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Per ulteriori informazioni, consulta [Running SQL Queries Using Amazon Athena nella Amazon Athena User Guide](#).

- Per API i dettagli, consulta Command [CreateNamedQueryReference](#) AWS CLI .

## create-work-group

Il seguente esempio di codice mostra come utilizzare `create-work-group`.

### AWS CLI

Per creare un gruppo di lavoro

L'`create-work-group` esempio seguente crea un gruppo di lavoro chiamato `Data_Analyst_Group` che ha la posizione di output dei risultati della query. `s3://awsdoc-example-bucket` Il comando crea un gruppo di lavoro che sostituisce le impostazioni di configurazione del client, che include la posizione di output dei risultati della query. Il comando abilita inoltre le CloudWatch metriche e aggiunge tre coppie di tag chiave-valore al gruppo di lavoro per distinguerlo dagli altri gruppi di lavoro. Si noti che l'`--configuration` argomento non ha spazi prima delle virgole che ne separano le opzioni.

```
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"
```

Questo comando non produce alcun output. Per vedere i risultati, usa `aws athena get-work-group --work-group Data_Analyst_Group`.

Per ulteriori informazioni, consulta [Managing Workgroups](#) nella Amazon Athena User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateWorkGroup](#) Reference.

## delete-data-catalog

Il seguente esempio di codice mostra come utilizzare `delete-data-catalog`.

### AWS CLI

Per eliminare un catalogo di dati

L'`delete-data-catalog` esempio seguente elimina il catalogo di `UnusedDataCatalog` dati.

```
aws athena delete-data-catalog \  
  --name UnusedDataCatalog
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di un catalogo: delete-data-catalog nella Guida per l'utente di Amazon Athena](#).

- Per API i dettagli, consulta AWS CLI Command [DeleteDataCatalog](#)Reference.

## delete-named-query

Il seguente esempio di codice mostra come utilizzare `delete-named-query`.

### AWS CLI

Per eliminare una query denominata

L'`delete-named-query`esempio seguente elimina la query denominata con l'ID specificato.

```
aws athena delete-named-query \  
  --named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Running SQL Queries Using Amazon Athena nella Amazon Athena User Guide](#).

- Per API i dettagli, consulta Command [DeleteNamedQuery](#)Reference AWS CLI .

## delete-work-group

Il seguente esempio di codice mostra come utilizzare `delete-work-group`.

### AWS CLI

Per eliminare un gruppo di lavoro

L'`delete-work-group`esempio seguente elimina il gruppo di lavoro. `TeamB`

```
aws athena delete-work-group \  
  --work-group TeamB
```



Questo comando non produce alcun output. Per confermare l'eliminazione, utilizzare `aws athena list-work-groups`

Per ulteriori informazioni, consulta [Managing Workgroups](#) nella Amazon Athena User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteWorkGroupReference](#).

## get-data-catalog

Il seguente esempio di codice mostra come utilizzare `get-data-catalog`.

### AWS CLI

Per restituire informazioni su un catalogo di dati

L'`get-data-catalog` seguente restituisce informazioni sul catalogo di `dynamo_db_catalog` dati.

```
aws athena get-data-catalog \  
  --name dynamo_db_catalog
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Mostrare i dettagli del catalogo: get-data-catalog](#) nella Guida per l'utente di Amazon Athena.

- Per API i dettagli, consulta [GetDataCatalog AWS CLI Command Reference](#).

## get-database

Il seguente esempio di codice mostra come utilizzare `get-database`.

### AWS CLI

Per restituire informazioni su un database in un catalogo di dati

L'`get-database` esempio seguente restituisce informazioni sul `sampledb` database nel catalogo `AwsDataCatalog` dati.

```
aws athena get-database \  
  --catalog-name AwsDataCatalog \  
  --database-name sampledb
```

Output:

```
{  
  "Database": {  
    "Name": "sampledb",  
    "Description": "Sample database",  
    "Parameters": {  
      "CreatedBy": "Athena",  
      "EXTERNAL": "TRUE"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta la sezione [Visualizzazione dei dettagli del database: `get-database`](#) nella Amazon Athena User Guide.

- Per API i dettagli, consulta Command [GetDatabaseReference](#) AWS CLI .

## get-named-query

Il seguente esempio di codice mostra come utilizzare `get-named-query`.

### AWS CLI

Per restituire una query denominata

L'`get-named-query` esempio seguente restituisce informazioni sulla query con l'ID specificato.

```
aws athena get-named-query \  
--named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Running SQL Queries Using Amazon Athena nella Amazon Athena User Guide](#).

- Per API i dettagli, consulta Command [GetNamedQuery](#)Reference AWS CLI .

## get-query-execution

Il seguente esempio di codice mostra come utilizzare `get-query-execution`.

AWS CLI

Per restituire informazioni sull'esecuzione di una query

L'`get-query-execution`esempio seguente restituisce informazioni sulla query con l'ID di query specificato.

```
aws athena get-query-execution \  
--query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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"
}
}

```

Per ulteriori informazioni, consulta [Running SQL Queries Using Amazon Athena nella Amazon Athena User Guide](#).

- Per API i dettagli, consulta Command [GetQueryExecution](#)Reference AWS CLI .

## get-query-results

Il seguente esempio di codice mostra come utilizzare `get-query-results`.

### AWS CLI

Per restituire i risultati di una query

L'`get-query-result`esempio seguente restituisce i risultati della query con l'ID di query specificato.

```
aws athena get-query-results \  
--query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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
}

```

Per ulteriori informazioni, consulta [Lavorare con i risultati delle query, i file di output e la cronologia delle query](#) nella Guida per l'utente di Amazon Athena.

- Per API i dettagli, consulta [GetQueryResults AWS CLI Command Reference](#).

## get-table-metadata

Il seguente esempio di codice mostra come utilizzare `get-table-metadata`.

### AWS CLI

Per restituire informazioni sui metadati relative a una tabella

L'esempio seguente restituisce le informazioni sui metadati relative alla `counties` tabella, inclusi i nomi delle colonne e i relativi tipi di dati, dal `sampledb` database del catalogo dati. `AwsDataCatalog`

```
aws athena get-table-metadata \
```

```
--catalog-name AwsDataCatalog \  
--database-name sampledb \  
--table-name counties
```

## Output:

```
{  
  "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"
      }
    }
  }
```

Per ulteriori informazioni, consulta [Mostrare i dettagli della tabella: get-table-metadata](#) nella Guida per l'utente di Amazon Athena.

- Per API i dettagli, consulta [GetTableMetadata AWS CLI Command Reference](#).

## get-work-group

Il seguente esempio di codice mostra come utilizzare `get-work-group`.

### AWS CLI

Per restituire informazioni su un gruppo di lavoro

L'`get-work-group` esempio seguente restituisce informazioni sul gruppo di AthenaAdmin lavoro.

```
aws athena get-work-group \  
  --work-group AthenaAdmin
```

Output:

```
{  
  "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  
  }  
}
```

```
}
```

Per ulteriori informazioni, consulta [Managing Workgroups](#) nella Amazon Athena User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetWorkGroup](#)Reference.

## list-data-catalogs

Il seguente esempio di codice mostra come utilizzare `list-data-catalogs`.

### AWS CLI

Per elencare i cataloghi di dati registrati con Athena

L'`list-data-catalogs`esempio seguente elenca i cataloghi di dati registrati con Athena.

```
aws athena list-data-catalogs
```

Output:

```
{
  "DataCatalogsSummary": [
    {
      "CatalogName": "AwsDataCatalog",
      "Type": "GLUE"
    },
    {
      "CatalogName": "cw_logs_catalog",
      "Type": "LAMBDA"
    },
    {
      "CatalogName": "cw_metrics_catalog",
      "Type": "LAMBDA"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Pubblicare cataloghi registrati: list-data-catalogs](#) nella Guida per l'utente di Amazon Athena.

- Per API i dettagli, consulta AWS CLI Command [ListDataCatalogs](#)Reference.

## list-databases

Il seguente esempio di codice mostra come utilizzare `list-databases`.

### AWS CLI

Per elencare i database in un catalogo di dati

L'`list-databases` esempio seguente elenca i database del catalogo `AwsDataCatalog` dati.

```
aws athena list-databases \  
  --catalog-name AwsDataCatalog
```

Output:

```
{  
  "DatabaseList": [  
    {  
      "Name": "default"  
    },  
    {  
      "Name": "mydatabase"  
    },  
    {  
      "Name": "newdb"  
    },  
    {  
      "Name": "sampledb",  
      "Description": "Sample database",  
      "Parameters": {  
        "CreatedBy": "Athena",  
        "EXTERNAL": "TRUE"  
      }  
    },  
    {  
      "Name": "webdata"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Listing Databases in a Catalog: list-databases](#) in the Amazon Athena User Guide.

- Per API i dettagli, consulta Command [ListDatabases](#)Reference AWS CLI .

## **list-named-queries**

Il seguente esempio di codice mostra come utilizzare `list-named-queries`.

AWS CLI

Per elencare le interrogazioni denominate per un gruppo di lavoro

L'`list-named-queries`esempio seguente elenca le interrogazioni denominate per il gruppo di lavoro. `AthenaAdmin`

```
aws athena list-named-queries \  
  --work-group AthenaAdmin
```

Output:

```
{  
  "NamedQueryIds": [  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"  
  ]  
}
```

Per ulteriori informazioni, consulta [Running SQL Queries Using Amazon Athena nella Amazon Athena User Guide](#).

- Per API i dettagli, consulta Command [ListNamedQueries](#)Reference AWS CLI .

## **list-query-executions**

Il seguente esempio di codice mostra come utilizzare `list-query-executions`.

AWS CLI

Per elencare l'interrogazione IDs delle interrogazioni in un gruppo di lavoro specificato

L'`list-query-executions`esempio seguente elenca un massimo di dieci interrogazioni IDs nel gruppo di lavoro. `AthenaAdmin`



```
aws athena list-query-executions \  
  --work-group AthenaAdmin \  
  --max-items 10
```

Output:

```
{  
  "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="
```

Per ulteriori informazioni, consulta [Lavorare con i risultati delle query, i file di output e la cronologia delle query](#) nella Guida per l'utente di Amazon Athena.

- Per API i dettagli, consulta [ListQueryExecutions AWS CLI Command Reference](#).

## list-table-metadata

Il seguente esempio di codice mostra come utilizzare `list-table-metadata`.

### AWS CLI

Per elencare i metadati per le tabelle nel database specificato di un catalogo di dati

L'`list-table-metadata` esempio seguente restituisce informazioni sui metadati per un massimo di due tabelle nel geography database del catalogo `AwsDataCatalog` dati.

```
aws athena list-table-metadata \  
  --catalog-name AwsDataCatalog \  
  --database-name geography \  
  --max-items 2
```

## Output:

```
{
  "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",
            "Type": "string",
            "Comment": "country name"
        },
        {
            "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=="
}

```

Per ulteriori informazioni, consulta [Mostrare i metadati per tutte le tabelle in un database: list-table-metadata](#) nella Amazon Athena User Guide.

- Per API i dettagli, consulta AWS CLI Command [ListTableMetadata](#)Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Esempio 1: elencare i tag per un gruppo di lavoro

L'`list-tags-for-resource` esempio seguente elenca i tag per il gruppo di `Data_Analyst_Group` lavoro.

```

aws athena list-tags-for-resource \
  --resource-arn arn:aws:athena:us-west-2:111122223333:workgroup/
Data_Analyst_Group

```

**Output:**

```
{
  "Tags": [
    {
      "Key": "Division",
      "Value": "West"
    },
    {
      "Key": "Team",
      "Value": "Big Data"
    },
    {
      "Key": "Location",
      "Value": "Seattle"
    }
  ]
}
```

Esempio 2: per elencare i tag per un catalogo di dati

L'`list-tags-for-resources` seguente elenca i tag per il catalogo di `dynamo_db_catalog` dati.

```
aws athena list-tags-for-resource \
  --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/
dynamo_db_catalog
```

**Output:**

```
{
  "Tags": [
    {
      "Key": "Division",
      "Value": "Mountain"
    },
    {
      "Key": "Organization",
      "Value": "Retail"
    },
    {
      "Key": "Product_Line",
```

```
        "Value": "Shoes"
      },
      {
        "Key": "Location",
        "Value": "Denver"
      }
    ]
  }
}
```

Per ulteriori informazioni, consulta [Elencare i tag per una risorsa: list-tags-for-resource](#) nella Amazon Athena User Guide.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI](#) Command Reference.

## list-work-groups

Il seguente esempio di codice mostra come utilizzare `list-work-groups`.

### AWS CLI

Per elencare i gruppi di lavoro

L'`list-work-groups` esempio seguente elenca i gruppi di lavoro nell'account corrente.

```
aws athena list-work-groups
```

Output:

```
{
  "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
}
]
```

Per ulteriori informazioni, consulta [Managing Workgroups](#) nella Amazon Athena User Guide.

- Per API i dettagli, consulta AWS CLI Command [ListWorkGroups](#)Reference.

## start-query-execution

Il seguente esempio di codice mostra come utilizzare `start-query-execution`.

### AWS CLI

Esempio 1: eseguire un'interrogazione in un gruppo di lavoro sulla tabella specificata nel database e nel catalogo di dati specificati

L'`start-query-execution`esempio seguente utilizza il `AthenaAdmin` gruppo di lavoro per eseguire una query sulla `cloudfront_logs` tabella `cflogsdatabase` nel catalogo `AwsDataCatalog` dati.

```
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
```

Output:

```
{  
  "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Per ulteriori informazioni, consulta [Running SQL Queries Using Amazon Athena nella Amazon Athena User Guide](#).

Esempio 2: eseguire una query che utilizza un gruppo di lavoro specifico per creare un database nel catalogo di dati specificato

L'`start-query-execution`esempio seguente utilizza il `AthenaAdmin` gruppo di lavoro per creare il database `newdb` nel catalogo dati predefinito. `AwsDataCatalog`

```
aws athena start-query-execution \  
  --query-string "create database if not exists newdb" \  
  --work-group "AthenaAdmin"
```

Output:

```
{  
  "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11112"  
}
```

Per ulteriori informazioni, consulta [Running SQL Queries Using Amazon Athena nella Amazon Athena User Guide](#).

Esempio 3: eseguire una query che crea una vista su una tabella nel database e nel catalogo di dati specificati

L'`start-query-execution`esempio seguente utilizza un'`SELECT`istruzione sulla `cloudfront_logs` tabella in `cflogsdatabase` per creare la vista `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
```

Output:

```
{  
  "QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11113"  
}
```

Per ulteriori informazioni, consulta [Running SQL Queries Using Amazon Athena nella Amazon Athena User Guide](#).

- Per API i dettagli, consulta Command [StartQueryExecution](#)Reference AWS CLI .



## stop-query-execution

Il seguente esempio di codice mostra come utilizzare `stop-query-execution`.

### AWS CLI

Per interrompere un'interrogazione in esecuzione

L'`stop-query-execution` esempio seguente interrompe la query con l'ID di query specificato.

```
aws athena stop-query-execution \  
  --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Running SQL Queries Using Amazon Athena nella Amazon Athena User Guide](#).

- Per API i dettagli, consulta Command [StopQueryExecution](#) Reference AWS CLI .

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere un tag a una risorsa

L'`tag-resource` esempio seguente aggiunge tre tag al catalogo `dynamo_db_catalog` dati.

```
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
```

Questo comando non produce alcun output. Per vedere il risultato, usa `aws athena list-tags-for-resource --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/dynamo_db_catalog`.

Per ulteriori informazioni, consulta [Aggiungere tag a resource: tag-resource](#) nella Amazon Athena User Guide.

- Per API i dettagli, consulta Command [TagResource](#)Reference AWS CLI .

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da una risorsa

L'`untag-resource`esempio seguente rimuove le Focus chiavi `Specialization` and e i valori associati dalla risorsa del catalogo `dynamo_db_catalog` dati.

```
aws athena untag-resource \  
  --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/  
dynamo_db_catalog \  
  --tag-keys Specialization Focus
```

Questo comando non produce alcun output. Per vedere i risultati, utilizzate il `list-tags-for-resource` comando.

Per ulteriori informazioni, consulta [Rimuovere tag da una risorsa: untag-resource](#) nella Amazon Athena User Guide.

- Per i API dettagli, consulta [UntagResource](#)Command Reference.AWS CLI

## update-data-catalog

Il seguente esempio di codice mostra come utilizzare `update-data-catalog`.

### AWS CLI

Per aggiornare un catalogo di dati

L'`update-data-catalog`esempio seguente aggiorna la funzione Lambda e la descrizione del catalogo `cw_logs_catalog` dati.

```
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
```

Questo comando non produce alcun output. Per vedere il risultato, usa `aws athena get-data-catalog --name cw_logs_catalog`.

Per ulteriori informazioni, consulta [Aggiornamento di un catalogo: update-data-catalog](#) nella Guida per l'utente di Amazon Athena.

- Per API i dettagli, consulta [UpdateDataCatalog AWS CLI Command Reference](#).

## update-work-group

Il seguente esempio di codice mostra come utilizzare `update-work-group`.

### AWS CLI

Per aggiornare un gruppo di lavoro

L'esempio seguente disattiva il `Data_Analyst_Group` gruppo di lavoro. Gli utenti non possono eseguire o creare query nel gruppo di lavoro disattivato, ma possono comunque visualizzare le metriche, i controlli dei limiti di utilizzo dei dati, le impostazioni del gruppo di lavoro, la cronologia delle query e le query salvate.

```
aws athena update-work-group \  
  --work-group Data_Analyst_Group \  
  --state DISABLED
```

Questo comando non produce alcun output. Per verificare il cambiamento di stato, utilizzate `aws athena get-work-group --work-group Data_Analyst_Group` e controllate la proprietà nell'`State` output.

Per ulteriori informazioni, consulta [Managing Workgroups](#) nella Amazon Athena User Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateWorkGroup](#) Reference.

## Esempi di Auto Scaling utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando l'AWS Command Line Interface Auto Scaling.



```
--auto-scaling-group-name my-asg \  
--target-group-arns arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Elastic Load Balancing e Amazon Auto EC2 Scaling nella Amazon Auto EC2 Scaling User Guide](#).

- Per API i dettagli, consulta Command [AttachLoadBalancerTargetGroups](#) Reference AWS CLI .

## attach-load-balancers

Il seguente esempio di codice mostra come utilizzare `attach-load-balancers`.

AWS CLI

Per collegare un Classic Load Balancer a un gruppo Auto Scaling

Questo esempio collega il Classic Load Balancer specificato al gruppo Auto Scaling specificato.

```
aws autoscaling attach-load-balancers \  
--load-balancer-names my-load-balancer \  
--auto-scaling-group-name my-asg
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Elastic Load Balancing e Amazon Auto EC2 Scaling nella Amazon Auto EC2 Scaling User Guide](#).

- Per API i dettagli, consulta Command [AttachLoadBalancers](#) Reference AWS CLI .

## cancel-instance-refresh

Il seguente esempio di codice mostra come utilizzare `cancel-instance-refresh`.

AWS CLI

Per annullare l'aggiornamento di un'istanza

L'esempio seguente annulla un aggiornamento dell'istanza in corso per il gruppo Auto Scaling specificato.

```
aws autoscaling cancel-instance-refresh \  
  --auto-scaling-group-name my-asg
```

Output:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

Per ulteriori informazioni, consulta [Annullare l'aggiornamento di un'istanza](#) nella Amazon EC2 Auto Scaling User Guide.

- Per API i dettagli, consulta AWS CLI Command [CancelInstanceRefresh](#) Reference.

## complete-lifecycle-action

Il seguente esempio di codice mostra come utilizzare `complete-lifecycle-action`.

AWS CLI

Per completare l'azione del ciclo di vita

Questo esempio notifica ad Amazon EC2 Auto Scaling che l'azione del ciclo di vita specificata è completa in modo che possa completare l'avvio o la chiusura dell'istanza.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i riferimenti al [ciclo di vita di Amazon EC2 Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per i API dettagli, consulta Command Reference. [CompleteLifecycleAction](#) AWS CLI

## create-auto-scaling-group

Il seguente esempio di codice mostra come utilizzare `create-auto-scaling-group`.

## AWS CLI

### Esempio 1: creare un gruppo Auto Scaling

L'create-auto-scaling-groupesempio seguente crea un gruppo Auto Scaling in sottoreti in più zone di disponibilità all'interno di una regione. Le istanze vengono avviate con la versione predefinita del modello di avvio specificato. Tieni presente che le impostazioni predefinite vengono utilizzate per la maggior parte delle altre impostazioni, come le politiche di terminazione e la configurazione del controllo dello stato di salute.

```
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-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i [gruppi Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

### Esempio 2: per collegare un Application Load Balancer, un Network Load Balancer o un Gateway Load Balancer

Questo esempio specifica il gruppo target per un sistema ARN di bilanciamento del carico che supporta il traffico previsto. Il tipo di controllo dello stato specifica ELB che quando Elastic Load Balancing segnala un'istanza come non integra, il gruppo Auto Scaling la sostituisce. Il comando definisce anche un periodo di tolleranza di secondi per il controllo dello stato di salute. 600 Il periodo di tolleranza aiuta a prevenire la chiusura anticipata delle istanze appena avviate.

```
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-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Elastic Load Balancing e Amazon Auto EC2 Scaling nella Amazon Auto EC2 Scaling User Guide](#).

Esempio 3: per specificare un gruppo di posizionamento e utilizzare la versione più recente del modello di lancio

Questo esempio avvia le istanze in un gruppo di posizionamento all'interno di una singola zona di disponibilità. Questo può essere utile per gruppi a bassa latenza con carichi di lavoro. HPC Questo esempio specifica anche la dimensione minima, la dimensione massima e la capacità desiderata del gruppo.

```
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"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Placement groups](#) nella Amazon EC2 User Guide for Linux Instances.

Esempio 4: Per specificare un gruppo Auto Scaling a istanza singola e utilizzare una versione specifica del modello di avvio

Questo esempio crea un gruppo Auto Scaling con capacità minima e massima impostata per 1 imporre l'esecuzione di un'istanza. Il comando specifica anche la versione 1 di un modello di avvio in cui è specificato l'ID di un modello esistente. ENI Quando si utilizza un modello di avvio che specifica un file esistente ENI per eth0, è necessario specificare una zona di disponibilità per il gruppo Auto Scaling che corrisponda all'interfaccia di rete, senza specificare anche un ID di sottorete nella richiesta.

```
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 \  
  --desired-capacity 1 \  
  --placement-group my-placement-group \  
  --vpc-zone-identifier "subnet-6194ea3b"
```



```
--min-size 1 \  
--max-size 1 \  
--availability-zones us-west-2a
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i [gruppi Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 5: specificare una politica di terminazione diversa

Questo esempio crea un gruppo Auto Scaling utilizzando una configurazione di avvio e imposta la politica di terminazione per terminare prima le istanze più vecchie. Il comando applica inoltre un tag al gruppo e alle relative istanze, con una chiave di e un valore diRole. WebServer

```
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-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Working with Amazon EC2 Auto Scaling policy di terminazione](#) nella Amazon EC2Auto Scaling User Guide.

Esempio 6: specificare un hook per il ciclo di vita del lancio

Questo esempio crea un gruppo Auto Scaling con un lifecycle hook che supporta un'azione personalizzata all'avvio dell'istanza.

```
aws autoscaling create-auto-scaling-group \  
  --cli-input-json file://~/config.json
```

Contenuto del file: config.json

```
{
```

```

"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"
}]
}

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i riferimenti al [ciclo di vita di Amazon EC2 Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 7: per specificare un hook del ciclo di vita della terminazione

Questo esempio crea un gruppo Auto Scaling con un lifecycle hook che supporta un'azione personalizzata alla chiusura dell'istanza.

```

aws autoscaling create-auto-scaling-group \
  --cli-input-json file://~/config.json

```

Contenuto di 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"
}

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i riferimenti al [ciclo di vita di Amazon EC2 Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 8: specificare una politica di terminazione personalizzata

Questo esempio crea un gruppo Auto Scaling che specifica una policy di terminazione della funzione Lambda personalizzata che indica ad Amazon Auto EC2 Scaling su quali istanze è sicuro terminare su larga scala.

```

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"

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creazione di una politica di terminazione personalizzata con Lambda](#) nella Amazon Auto EC2 Scaling User Guide.

- Per API i dettagli, consulta Command [CreateAutoScalingGroupReference](#) AWS CLI .

## create-launch-configuration

Il seguente esempio di codice mostra come utilizzare `create-launch-configuration`.

### AWS CLI

Esempio 1: per creare una configurazione di avvio

Questo esempio crea una semplice configurazione di avvio.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creazione di una configurazione di avvio](#) nella Amazon EC2 Auto Scaling User Guide.

Esempio 2: creare una configurazione di avvio con un gruppo di sicurezza, una coppia di chiavi e uno script di avvio

Questo esempio crea una configurazione di avvio con un gruppo di sicurezza, una coppia di chiavi e uno script di bootstrapping contenuti nei dati utente.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creazione di una configurazione di avvio](#) nella Amazon EC2 Auto Scaling User Guide.

### Esempio 3: creare una configurazione di avvio con un ruolo IAM

Questo esempio crea una configurazione di avvio con il nome del profilo dell'istanza di un IAM ruolo.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --iam-instance-profile my-autoscaling-role
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [IAM il ruolo per le applicazioni eseguite su EC2 istanze Amazon](#) nella Amazon EC2 Auto Scaling User Guide.

### Esempio 4: creare una configurazione di avvio con il monitoraggio dettagliato abilitato

Questo esempio crea una configurazione di avvio con il monitoraggio EC2 dettagliato abilitato, che invia le EC2 metriche CloudWatch in periodi di 1 minuto.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --instance-monitoring Enabled=true
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configurazione del monitoraggio per le istanze Auto Scaling](#) nella Amazon Auto EC2Scaling User Guide.

### Esempio 5: creare una configurazione di avvio che avvii le istanze Spot

Questo esempio crea una configurazione di avvio che utilizza le istanze Spot come unica opzione di acquisto.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc \  
  --image-id ami-04d5cc9b88example \  
  --instance-type m5.large \  
  --spot-price LowestPrice
```

```
--spot-price "0.50"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Richiesta di istanze Spot](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 6: creare una configurazione di avvio utilizzando un'istanza EC2

Questo esempio crea una configurazione di avvio basata sugli attributi di un'istanza esistente. Sostituisce la locazione e l'eventuale impostazione di un indirizzo IP pubblico includendo le --placement-tenancy opzioni and. --no-associate-public-ip-address

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creazione di una configurazione di avvio utilizzando un'EC2istanza](#) nella Amazon EC2 Auto Scaling User Guide.

Esempio 7: creare una configurazione di avvio con una mappatura dei dispositivi a blocchi per un volume Amazon EBS

Questo esempio crea una configurazione di avvio con una mappatura dei dispositivi a blocchi per un EBS gp3 volume Amazon con il nome del dispositivo /dev/sdh e una dimensione del volume di 20.

```
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"}}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [EBS](#) Amazon EC2 Auto Scaling API Reference.

Per informazioni sulla sintassi per citare i valori dei parametri JSON -formatted, consulta [Usare le virgolette con le stringhe nella Guida per l'utente dell' AWS CLIinterfaccia a riga](#) di comando.AWS

Esempio 8: creare una configurazione di avvio con una mappatura dei dispositivi a blocchi per un volume di archiviazione di istanze

Questo esempio crea una configurazione di avvio con ephemeral1 come volume di archivio delle istanze con il nome /dev/sdc del dispositivo.

```
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"}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [BlockDeviceMapping](#)Amazon EC2 Auto Scaling API Reference.

Per informazioni sulla sintassi per citare i valori dei parametri JSON -formatted, consulta [Usare le virgolette con le stringhe nella Guida per l'utente dell' AWS CLIinterfaccia a riga](#) di comando.AWS

Esempio 9: creare una configurazione di avvio e impedire a un dispositivo di collegarsi al momento dell'avvio

Questo esempio crea una configurazione di avvio che sopprime un dispositivo a blocchi specificato dalla mappatura dei dispositivi a blocchi di AMI (ad esempio,). /dev/sdf

```
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":""}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [BlockDeviceMapping](#)Amazon EC2 Auto Scaling API Reference.

Per informazioni sulla sintassi per citare i valori dei parametri JSON -formatted, consulta [Usare le virgolette con le stringhe nella Guida per l'utente dell' AWS CLIinterfaccia a riga](#) di comando.AWS

- Per i API dettagli, vedere in Command Reference. [CreateLaunchConfiguration](#)AWS CLI

## create-or-update-tags

Il seguente esempio di codice mostra come utilizzare `create-or-update-tags`.

### AWS CLI

Per creare o aggiornare i tag per un gruppo di Auto Scaling

Questo esempio aggiunge due tag al gruppo Auto Scaling specificato.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Tagging di gruppi e istanze di Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per i API dettagli, consulta [CreateOrUpdateTags](#) Command Reference.AWS CLI

## delete-auto-scaling-group

Il seguente esempio di codice mostra come utilizzare `delete-auto-scaling-group`.

### AWS CLI

Esempio 1: Per eliminare il gruppo Auto Scaling specificato

Questo esempio elimina il gruppo Auto Scaling specificato.

```
aws autoscaling delete-auto-scaling-group \  
  --auto-scaling-group-name my-asg
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione dell'infrastruttura Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 2: Per forzare l'eliminazione del gruppo Auto Scaling specificato



Per eliminare il gruppo Auto Scaling senza attendere che le istanze del gruppo terminino, utilizzate l'opzione. `--force-delete`

```
aws autoscaling delete-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --force-delete
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione dell'infrastruttura Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per API i dettagli, consulta Command [DeleteAutoScalingGroup](#)Reference AWS CLI .

## **delete-launch-configuration**

Il seguente esempio di codice mostra come utilizzare `delete-launch-configuration`.

AWS CLI

Per eliminare una configurazione di avvio

Questo esempio elimina la configurazione di avvio specificata.

```
aws autoscaling delete-launch-configuration \  
  --launch-configuration-name my-launch-config
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione dell'infrastruttura Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per API i dettagli, consulta Command [DeleteLaunchConfiguration](#)Reference AWS CLI .

## **delete-lifecycle-hook**

Il seguente esempio di codice mostra come utilizzare `delete-lifecycle-hook`.

AWS CLI

Per eliminare un hook del ciclo di vita

Questo esempio elimina l'hook del ciclo di vita specificato.

```
aws autoscaling delete-lifecycle-hook \  
  --lifecycle-hook-name my-lifecycle-hook \  
  --auto-scaling-group-name my-asg
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteLifecycleHook](#) in AWS CLI Command Reference.

## delete-notification-configuration

Il seguente esempio di codice mostra come utilizzare `delete-notification-configuration`.

AWS CLI

Per eliminare una notifica di Auto Scaling

Questo esempio elimina la notifica specificata dal gruppo Auto Scaling specificato.

```
aws autoscaling delete-notification-configuration \  
  --auto-scaling-group-name my-asg \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminare la configurazione delle notifiche](#) nella Amazon EC2 Auto Scaling User Guide.

- Per API i dettagli, consulta [DeleteNotificationConfiguration AWS CLI Command Reference](#).

## delete-policy

Il seguente esempio di codice mostra come utilizzare `delete-policy`.

AWS CLI

Per eliminare una politica di scalabilità

Questo esempio elimina la politica di scalabilità specificata.

```
aws autoscaling delete-policy \  
  --auto-scaling-group-name my-asg \  
  --policy-name my-policy
```

```
--policy-name alb1000-target-tracking-scaling-policy
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeletePolicy](#) in AWS CLI Command Reference.

## delete-scheduled-action

Il seguente esempio di codice mostra come utilizzare `delete-scheduled-action`.

### AWS CLI

Per eliminare un'azione pianificata da un gruppo Auto Scaling

Questo esempio elimina l'azione pianificata specificata dal gruppo Auto Scaling specificato.

```
aws autoscaling delete-scheduled-action \  
  --auto-scaling-group-name my-asg \  
  --scheduled-action-name my-scheduled-action
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteScheduledAction](#) in AWS CLI Command Reference.

## delete-tags

Il seguente esempio di codice mostra come utilizzare `delete-tags`.

### AWS CLI

Per eliminare un tag da un gruppo Auto Scaling

Questo esempio elimina il tag specificato dal gruppo Auto Scaling specificato.

```
aws autoscaling delete-tags \  
  --tags ResourceId=my-asg,ResourceType=auto-scaling-group,Key=Dept,Value=Research
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Tagging di gruppi e istanze di Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per i API dettagli, consulta [DeleteTags](#) Command Reference.AWS CLI

## delete-warm-pool

Il seguente esempio di codice mostra come utilizzare `delete-warm-pool`.

### AWS CLI

Esempio 1: Per eliminare una piscina calda

L'esempio seguente elimina il pool caldo per il gruppo Auto Scaling specificato.

```
aws autoscaling delete-warm-pool \  
  --auto-scaling-group-name my-asg
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Warm pools for Amazon EC2 Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 2: forzare l'eliminazione di una piscina calda

Per eliminare la piscina calda senza attendere che le relative istanze vengano terminate, utilizzate l'`--force-delete` opzione.

```
aws autoscaling delete-warm-pool \  
  --auto-scaling-group-name my-asg \  
  --force-delete
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Warm pools for Amazon EC2 Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteWarmPool](#) Reference.

## describe-account-limits

Il seguente esempio di codice mostra come utilizzare `describe-account-limits`.

### AWS CLI

Per descrivere i limiti del tuo EC2 account Amazon Auto Scaling

Questo esempio descrive i limiti di Amazon EC2 Auto Scaling per il tuo AWS account.

```
aws autoscaling describe-account-limits
```

Output:

```
{
  "NumberOfLaunchConfigurations": 5,
  "MaxNumberOfLaunchConfigurations": 100,
  "NumberOfAutoScalingGroups": 3,
  "MaxNumberOfAutoScalingGroups": 20
}
```

Per ulteriori informazioni, consulta le [quote del servizio Amazon EC2 Auto Scaling](#) nella Amazon EC2 Auto Scaling User Guide.

- Per API i dettagli, consulta Command [DescribeAccountLimits](#) Reference AWS CLI .

## describe-adjustment-types

Il seguente esempio di codice mostra come utilizzare `describe-adjustment-types`.

AWS CLI

Per descrivere i tipi di regolazione della scala disponibili

Questo esempio descrive i tipi di regolazione disponibili.

```
aws autoscaling describe-adjustment-types
```

Output:

```
{
  "AdjustmentTypes": [
    {
      "AdjustmentType": "ChangeInCapacity"
    },
    {
      "AdjustmentType": "ExactCapacity"
    },
    {
      "AdjustmentType": "PercentChangeInCapacity"
    }
  ]
}
```

```
]
}
```

Per ulteriori informazioni, consulta i [tipi di regolazione della scalabilità](#) nella Guida per l'utente di Amazon EC2 Auto Scaling.

- Per API i dettagli, consulta AWS CLI Command [DescribeAdjustmentTypes](#) Reference.

## describe-auto-scaling-groups

Il seguente esempio di codice mostra come utilizzare `describe-auto-scaling-groups`.

### AWS CLI

Esempio 1: Per descrivere il gruppo Auto Scaling specificato

Questo esempio descrive il gruppo Auto Scaling specificato.

```
aws autoscaling describe-auto-scaling-groups \
  --auto-scaling-group-name my-asg
```

Output:

```
{
  "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": []
    }
]
}

```

Esempio 2: Per descrivere i primi 100 gruppi Auto Scaling specificati

Questo esempio descrive i gruppi Auto Scaling specificati. Consente di specificare fino a 100 nomi di gruppi.

```

aws autoscaling describe-auto-scaling-groups \
  --max-items 100 \

```

```
--auto-scaling-group-name "group1" "group2" "group3" "group4"
```

Vedi l'esempio 1 per un esempio di output.

Esempio 3: Per descrivere un gruppo Auto Scaling nella regione specificata

Questo esempio descrive i gruppi Auto Scaling nella regione specificata, fino a un massimo di 75 gruppi.

```
aws autoscaling describe-auto-scaling-groups \  
--max-items 75 \  
--region us-east-1
```

Vedi l'esempio 1 per un esempio di output.

Esempio 4: Per descrivere il numero specificato del gruppo Auto Scaling

Per restituire un numero specifico di gruppi Auto Scaling, utilizzate l'`--max-items` opzione.

```
aws autoscaling describe-auto-scaling-groups \  
--max-items 1
```

Vedi l'esempio 1 per un esempio di output.

Se l'output include un `NextToken` campo, ci sono più gruppi. Per ottenere i gruppi aggiuntivi, utilizzate il valore di questo campo con l'`--starting-token` opzione in una chiamata successiva come segue.

```
aws autoscaling describe-auto-scaling-groups \  
--starting-token Z3M3LMPEXAMPLE
```

Vedi l'esempio 1 per un esempio di output.

Esempio 5: Per descrivere i gruppi di Auto Scaling che utilizzano configurazioni di avvio

Questo esempio utilizza l'`--query` opzione per descrivere i gruppi di Auto Scaling che utilizzano configurazioni di avvio.

```
aws autoscaling describe-auto-scaling-groups \  
--query 'AutoScalingGroups[?LaunchConfigurationName!=`null`]'
```



## Output:

```
[
  {
    "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": []
  }
]
```

```
}  
]
```

Per ulteriori informazioni, consulta [Filtrare l' AWS CLI output nella Guida](#) per l'utente dell'interfaccia a riga di AWS comando.

- Per API i dettagli, vedere [DescribeAutoScalingGroups](#) in AWS CLI Command Reference.

## describe-auto-scaling-instances

Il seguente esempio di codice mostra come utilizzare `describe-auto-scaling-instances`.

### AWS CLI

Esempio 1: Per descrivere una o più istanze

Questo esempio descrive l'istanza specificata.

```
aws autoscaling describe-auto-scaling-instances \  
  --instance-ids i-06905f55584de02da
```

Output:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

Esempio 2: Per descrivere una o più istanze

Questo esempio utilizza l'`--max-items` opzione per specificare quante istanze restituire con questa chiamata.

```
aws autoscaling describe-auto-scaling-instances \  
  --max-items 1
```

Se l'output include un `NextToken` campo, ci sono più istanze. Per ottenere le istanze aggiuntive, utilizzate il valore di questo campo con l'`--starting-token` opzione in una chiamata successiva come segue.

```
aws autoscaling describe-auto-scaling-instances \  
  --starting-token Z3M3LMPEXAMPLE
```

Vedi l'esempio 1 per un output di esempio.

Esempio 3: Per descrivere le istanze che utilizzano configurazioni di avvio

Questo esempio utilizza l'`--query` opzione per descrivere le istanze che utilizzano configurazioni di avvio.

```
aws autoscaling describe-auto-scaling-instances \  
  --query 'AutoScalingInstances[?LaunchConfigurationName!=`null`]'
```

Output:

```
[  
  {  
    "InstanceId": "i-088c57934a6449037",  
    "InstanceType": "t2.micro",  
    "AutoScalingGroupName": "my-asg",  
    "AvailabilityZone": "us-west-2c",  
    "LifecycleState": "InService",  
    "HealthStatus": "HEALTHY",  
    "LaunchConfigurationName": "my-lc",  
    "ProtectedFromScaleIn": false  
  }  
]
```

Per ulteriori informazioni, consulta [Filtrare l' AWS CLI output nella Guida](#) per l'utente dell'interfaccia a riga di AWS comando.

- Per API i dettagli, vedere [DescribeAutoScalingInstances](#) in AWS CLI Command Reference.

## describe-auto-scaling-notification-types

Il seguente esempio di codice mostra come utilizzare `describe-auto-scaling-notification-types`.

### AWS CLI

Per descrivere i tipi di notifica disponibili

Questo esempio descrive i tipi di notifica disponibili.

```
aws autoscaling describe-auto-scaling-notification-types
```

Output:

```
{
  "AutoScalingNotificationTypes": [
    "autoscaling:EC2_INSTANCE_LAUNCH",
    "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
    "autoscaling:EC2_INSTANCE_TERMINATE",
    "autoscaling:EC2_INSTANCE_TERMINATE_ERROR",
    "autoscaling:TEST_NOTIFICATION"
  ]
}
```

Per ulteriori informazioni, consulta [Ricevere SNS notifiche Amazon quando il tuo gruppo Auto Scaling aumenta nella Amazon EC2 Auto Scaling User Guide](#).

- Per API i dettagli, consulta Command [DescribeAutoScalingNotificationTypes](#) Reference AWS CLI .

## describe-instance-refreshes

Il seguente esempio di codice mostra come utilizzare `describe-instance-refreshes`.

### AWS CLI

Per descrivere gli aggiornamenti delle istanze

L'`describe-instance-refreshes` seguente restituisce una descrizione di tutte le richieste di aggiornamento delle istanze per il gruppo Auto Scaling specificato, incluso il messaggio di stato e (se disponibile) il motivo dello stato.

```
aws autoscaling describe-instance-refreshes \  
--auto-scaling-group-name my-asg
```

Output:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

```

    }
  }
]
}

```

Per ulteriori informazioni, consulta [Verifica dello stato di un aggiornamento dell'istanza](#) nella Amazon EC2 Auto Scaling User Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeInstanceRefreshes](#) Reference.

## describe-launch-configurations

Il seguente esempio di codice mostra come utilizzare `describe-launch-configurations`.

### AWS CLI

Esempio 1: Per descrivere la configurazione di avvio specificata

Questo esempio descrive la configurazione di avvio specificata.

```

aws autoscaling describe-launch-configurations \
  --launch-configuration-names my-launch-config

```

Output:

```

{
  "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"
    }
  }
]
}

```

Esempio 2: Per descrivere un numero specificato di configurazioni di avvio

Per restituire un numero specifico di configurazioni di avvio, usa l' `--max-items` opzione.

```

aws autoscaling describe-launch-configurations \
  --max-items 1

```

Se l'output include un `NextToken` campo, ci sono più configurazioni di avvio. Per ottenere le configurazioni di avvio aggiuntive, utilizzate il valore di questo campo con l' `--starting-token` opzione in una chiamata successiva come segue.

```

aws autoscaling describe-launch-configurations \
  --starting-token Z3M3LMPEXAMPLE

```

- Per API i dettagli, vedere [DescribeLaunchConfigurations](#) in AWS CLI Command Reference.

## describe-lifecycle-hook-types

Il seguente esempio di codice mostra come utilizzare `describe-lifecycle-hook-types`.

### AWS CLI

Per descrivere i tipi di hook del ciclo di vita disponibili

Questo esempio descrive i tipi di hook del ciclo di vita disponibili.

```
aws autoscaling describe-lifecycle-hook-types
```

Output:

```
{
  "LifecycleHookTypes": [
    "autoscaling:EC2_INSTANCE_LAUNCHING",
    "autoscaling:EC2_INSTANCE_TERMINATING"
  ]
}
```

- Per API i dettagli, vedere [DescribeLifecycleHookTypes](#) in AWS CLI Command Reference.

## describe-lifecycle-hooks

Il seguente esempio di codice mostra come utilizzare `describe-lifecycle-hooks`.

### AWS CLI

Per descrivere i ganci del ciclo di vita

Questo esempio descrive gli hook del ciclo di vita per il gruppo Auto Scaling specificato.

```
aws autoscaling describe-lifecycle-hooks \
  --auto-scaling-group-name my-asg
```

Output:

```
{
  "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"
    }
]
}

```

- Per i API dettagli, vedere [DescribeLifecycleHooks](#) in Command Reference.AWS CLI

## describe-load-balancer-target-groups

Il seguente esempio di codice mostra come utilizzare `describe-load-balancer-target-groups`.

### AWS CLI

Per descrivere i gruppi target dei sistemi di bilanciamento del carico per un gruppo di Auto Scaling

Questo esempio descrive i gruppi target del sistema di bilanciamento del carico collegati al gruppo Auto Scaling specificato.

```
aws autoscaling describe-load-balancer-target-groups \
  --auto-scaling-group-name my-asg
```

### Output:

```

{
  "LoadBalancerTargetGroups": [
    {
      "LoadBalancerTargetGroupARN": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
      "State": "Added"
    }
  ]
}

```

```
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeLoadBalancerTargetGroups](#) in AWS CLI Command Reference.

## describe-load-balancers

Il seguente esempio di codice mostra come utilizzare `describe-load-balancers`.

### AWS CLI

Per descrivere il gruppo Classic Load Balancer for an Auto Scaling

Questo esempio descrive i Classic Load Balancer per il gruppo Auto Scaling specificato.

```
aws autoscaling describe-load-balancers \
  --auto-scaling-group-name my-asg
```

Output:

```
{
  "LoadBalancers": [
    {
      "State": "Added",
      "LoadBalancerName": "my-load-balancer"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeLoadBalancers](#) in AWS CLI Command Reference.

## describe-metric-collection-types

Il seguente esempio di codice mostra come utilizzare `describe-metric-collection-types`.

### AWS CLI

Per descrivere i tipi di raccolta di metriche disponibili

Questo esempio descrive i tipi di raccolta di metriche disponibili.

```
aws autoscaling describe-metric-collection-types
```

Output:

```
{
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta le [metriche del gruppo Auto Scaling](#) nella Amazon EC2 Auto Scaling User Guide.

- Per API i dettagli, consulta Command [DescribeMetricCollectionTypes](#) Reference AWS CLI .

## describe-notification-configurations

Il seguente esempio di codice mostra come utilizzare `describe-notification-configurations`.

### AWS CLI

Esempio 1: Per descrivere le configurazioni di notifica di un gruppo specificato

Questo esempio descrive le configurazioni di notifica per il gruppo Auto Scaling specificato.

```

aws autoscaling describe-notification-configurations \
  --auto-scaling-group-name my-asg

```

Output:

```

{
  "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"
    }
  ]
}

```

```
]
}
```

Per ulteriori informazioni, consulta [Ricevere SNS notifiche Amazon quando il tuo gruppo Auto Scaling aumenta nella Amazon EC2Auto Scaling User Guide](#).

Esempio 1: per descrivere un numero specifico di configurazioni di notifica

Per restituire un numero specifico di configurazioni di notifica, utilizzare il `max-items` parametro.

```
aws autoscaling describe-notification-configurations \
  --auto-scaling-group-name my-auto-scaling-group \
  --max-items 1
```

Output:

```
{
  "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"
    }
  ]
}
```

Se l'output include un `NextToken` campo, ci sono più configurazioni di notifica. Per ottenere le configurazioni di notifica aggiuntive, utilizzate il valore di questo campo con il `starting-token` parametro in una chiamata successiva come segue.

```
aws autoscaling describe-notification-configurations \
  --auto-scaling-group-name my-asg \
  --starting-token Z3M3LMPEXAMPLE
```

Per ulteriori informazioni, consulta [Ricevere SNS notifiche Amazon quando il tuo gruppo Auto Scaling aumenta nella Amazon EC2Auto Scaling User Guide](#).

- Per API i dettagli, consulta Command [DescribeNotificationConfigurations](#) Reference AWS CLI .

## describe-policies

Il seguente esempio di codice mostra come utilizzare `describe-policies`.

### AWS CLI

Esempio 1: Per descrivere le politiche di scalabilità di un gruppo specificato

Questo esempio descrive le politiche di ridimensionamento per il gruppo Auto Scaling specificato.

```
aws autoscaling describe-policies \  
  --auto-scaling-group-name my-asg
```

Output:

```
{  
  "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
      }
    ]
  }

```

Per ulteriori informazioni, consulta [Dynamic Scaling](#) nella Amazon EC2 Auto Scaling User Guide.

Esempio 2: Per descrivere le politiche di scalabilità di un nome specifico

Per restituire politiche di ridimensionamento specifiche, utilizzare l'`--policy-names` opzione.

```

aws autoscaling describe-policies \
  --auto-scaling-group-name my-asg \
  --policy-names cpu40-target-tracking-scaling-policy

```

Vedi l'esempio 1 per un esempio di output.

Per ulteriori informazioni, consulta [Dynamic Scaling](#) nella Amazon EC2 Auto Scaling User Guide.

Esempio 3: Per descrivere una serie di politiche di scalabilità

Per restituire un numero specifico di politiche, utilizzare l'`--max-items` opzione.

```

aws autoscaling describe-policies \
  --auto-scaling-group-name my-asg \
  --max-items 1

```

Vedi l'esempio 1 per un esempio di output.

Se l'output include un `NextToken` campo, utilizza il valore di questo campo con l'`--starting-token` opzione in una chiamata successiva per ottenere le politiche aggiuntive.

```

aws autoscaling describe-policies --auto-scaling-group-name my-asg --starting-
token Z3M3LMPEXAMPLE

```

Per ulteriori informazioni, consulta [Dynamic Scaling](#) nella Amazon EC2 Auto Scaling User Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribePolicies](#) Reference.

## describe-scaling-activities

Il seguente esempio di codice mostra come utilizzare `describe-scaling-activities`.



## AWS CLI

Esempio 1: Per descrivere le attività di ridimensionamento per il gruppo specificato

Questo esempio descrive le attività di ridimensionamento per il gruppo Auto Scaling specificato.

```
aws autoscaling describe-scaling-activities \  
  --auto-scaling-group-name my-asg
```

Output:

```
{  
  "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\"}"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Verificare un'attività di scaling per un gruppo di Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 2: per descrivere le attività di ridimensionamento per un gruppo eliminato

Per descrivere le attività di ridimensionamento dopo l'eliminazione del gruppo Auto Scaling, aggiungete `--include-deleted-groups` l'opzione.

```
aws autoscaling describe-scaling-activities \  
  --auto-scaling-group-name my-asg \  
  --include-deleted-groups
```

**--include-deleted-groups**

## Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione [Risoluzione dei problemi di Amazon Auto EC2 Scaling nella Amazon Auto EC2 Scaling User Guide](#).

Esempio 3: Per descrivere un numero specifico di attività di scalabilità

Per restituire un numero specifico di attività, utilizzare l' `--max-items` opzione.

```
aws autoscaling describe-scaling-activities \
```

```
--max-items 1
```

Output:

```
{
  "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\\\"}"
    }
  ]
}
```

Se l'output include un `NextToken` campo, ci sono più attività. Per ottenere le attività aggiuntive, usa il valore di questo campo con l'`--starting-token` opzione in una chiamata successiva come segue.

```
aws autoscaling describe-scaling-activities \
  --starting-token Z3M3LMPEXAMPLE
```

Per ulteriori informazioni, consulta [Verificare un'attività di scaling per un gruppo di Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per API i dettagli, consulta Command [DescribeScalingActivities](#) Reference AWS CLI .

## describe-scaling-process-types

Il seguente esempio di codice mostra come utilizzare `describe-scaling-process-types`.

## AWS CLI

Per descrivere i tipi di processo disponibili

Questo esempio descrive i tipi di processo disponibili.

```
aws autoscaling describe-scaling-process-types
```

Output:

```
{
  "Processes": [
    {
      "ProcessName": "AZRebalance"
    },
    {
      "ProcessName": "AddToLoadBalancer"
    },
    {
      "ProcessName": "AlarmNotification"
    },
    {
      "ProcessName": "HealthCheck"
    },
    {
      "ProcessName": "InstanceRefresh"
    },
    {
      "ProcessName": "Launch"
    },
    {
      "ProcessName": "ReplaceUnhealthy"
    },
    {
      "ProcessName": "ScheduledActions"
    },
    {
      "ProcessName": "Terminate"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Sospensione e ripresa dei processi di scalabilità nella Amazon Auto Scaling](#) User Guide. EC2

- Per i API dettagli, consulta Command Reference. [DescribeScalingProcessTypes](#) AWS CLI

## describe-scheduled-actions

Il seguente esempio di codice mostra come utilizzare `describe-scheduled-actions`.

### AWS CLI

Esempio 1: Per descrivere tutte le azioni pianificate

Questo esempio descrive tutte le azioni pianificate.

```
aws autoscaling describe-scheduled-actions
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Scheduled Scaling](#) nella Amazon EC2 Auto Scaling User Guide.

Esempio 2: Per descrivere le azioni pianificate per il gruppo specificato

Per descrivere le azioni pianificate per uno specifico gruppo di Auto Scaling, utilizzate l'`--auto-scaling-group-name` opzione.

```
aws autoscaling describe-scheduled-actions \  
  --auto-scaling-group-name my-asg
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Scheduled Scaling](#) nella Amazon EC2 Auto Scaling User Guide.

Esempio 3: Per descrivere l'azione pianificata specificata

Per descrivere un'azione pianificata specifica, utilizzare l'`--scheduled-action-names` opzione.

```
aws autoscaling describe-scheduled-actions \  
  --scheduled-action-names my-recurring-action
```

Output:

```
{  
  "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"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Scheduled Scaling](#) nella Amazon EC2 Auto Scaling User Guide.

Esempio 4: Per descrivere le azioni pianificate con un'ora di inizio specificata

Per descrivere le azioni pianificate che iniziano in un'ora specifica, utilizza l'`--start-time` opzione.

```

aws autoscaling describe-scheduled-actions \
  --start-time "2023-12-01T04:00:00Z"

```

Output:

```

{
  "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"  
    }  
]  
}
```

Per ulteriori informazioni, consulta [Scheduled Scaling](#) nella Amazon EC2 Auto Scaling User Guide.

Esempio 5: Per descrivere le azioni pianificate che terminano in un momento specifico

Per descrivere le azioni pianificate che terminano in un momento specifico, utilizza l'`--end-time` opzione.

```
aws autoscaling describe-scheduled-actions \  
  --end-time "2023-12-01T04:00:00Z"
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Scheduled Scaling](#) nella Amazon EC2 Auto Scaling User Guide.



## Esempio 6: Per descrivere un numero specifico di azioni pianificate

Per restituire un numero specifico di azioni pianificate, utilizzare l' `--max-items` opzione.

```
aws autoscaling describe-scheduled-actions \  
  --auto-scaling-group-name my-asg \  
  --max-items 1
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Se l'output include un `NextToken` campo, ci sono più azioni pianificate. Per ottenere le azioni pianificate aggiuntive, utilizza il valore di questo campo con l' `--starting-token` opzione in una chiamata successiva come segue.

```
aws autoscaling describe-scheduled-actions \  
  --auto-scaling-group-name my-asg \  
  --starting-token Z3M3LMPEXAMPLE
```

Per ulteriori informazioni, consulta [Scheduled Scaling](#) nella Amazon EC2 Auto Scaling User Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeScheduledActions](#) Reference.

## describe-tags

Il seguente esempio di codice mostra come utilizzare `describe-tags`.

### AWS CLI

Per descrivere tutti i tag

Questo esempio descrive tutti i tag.

```
aws autoscaling describe-tags
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione [Tagging di gruppi e istanze di Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 2: per descrivere i tag per un gruppo specifico

Per descrivere i tag per uno specifico gruppo di Auto Scaling, utilizzate l'`--filters` opzione.

```
aws autoscaling describe-tags --filters Name=auto-scaling-group,Values=my-asg
```

Per ulteriori informazioni, consulta la sezione [Tagging di gruppi e istanze di Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 3: per descrivere il numero di tag specificato

Per restituire un numero specifico di tag, usa l'`--max-items` opzione.

```
aws autoscaling describe-tags \  
  --max-items 1
```

Se l'output include un `NextToken` campo, ci sono più tag. Per ottenere i tag aggiuntivi, usa il valore di questo campo con l'`--starting-token` opzione in una chiamata successiva come segue.

```
aws autoscaling describe-tags \  
  --filters Name=auto-scaling-group,Values=my-asg \  
  --starting-token Z3M3LMPEXAMPLE
```

Per ulteriori informazioni, consulta la sezione [Tagging di gruppi e istanze di Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per i API dettagli, consulta [DescribeTags](#) Command Reference.AWS CLI

## describe-termination-policy-types

Il seguente esempio di codice mostra come utilizzare `describe-termination-policy-types`.

AWS CLI

Per descrivere i tipi di policy di terminazione disponibili

Questo esempio descrive i tipi di policy di terminazione disponibili.

```
aws autoscaling describe-termination-policy-types
```

Output:

```
{  
  "TerminationPolicyTypes": [  
    "AllocationStrategy",  
    "ClosestToNextInstanceHour",
```

```
    "Default",
    "NewestInstance",
    "OldestInstance",
    "OldestLaunchConfiguration",
    "OldestLaunchTemplate"
  ]
}
```

Per ulteriori informazioni, consulta la sezione [Controllo delle istanze di Auto Scaling che terminano durante la scalabilità nella](#) Amazon Auto Scaling EC2 User Guide.

- Per i API dettagli, consulta [DescribeTerminationPolicyTypes](#) Command Reference.AWS CLI

## describe-warm-pool

Il seguente esempio di codice mostra come utilizzare `describe-warm-pool`.

### AWS CLI

Per descrivere una piscina calda

Questo esempio descrive il pool caldo per il gruppo Auto Scaling specificato.

```
aws autoscaling describe-warm-pool \
  --auto-scaling-group-name my-asg
```

Output:

```
{
  "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"
      }
    }
  ]
}

```

Per ulteriori informazioni, consulta [Warm pools for Amazon EC2 Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeWarmPool](#) Reference.

## detach-instances

Il seguente esempio di codice mostra come utilizzare `detach-instances`.

### AWS CLI

Per scollegare un'istanza da un gruppo Auto Scaling

Questo esempio scollega l'istanza specificata dal gruppo Auto Scaling specificato.

```

aws autoscaling detach-instances \
  --instance-ids i-030017cfa84b20135 \
  --auto-scaling-group-name my-asg \
  --should-decrement-desired-capacity

```

Output:

```

{
  "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\\\"}"
  }
]
}

```

- Per API i dettagli, vedere [DetachInstances](#) in AWS CLI Command Reference.

## detach-load-balancer-target-groups

Il seguente esempio di codice mostra come utilizzare `detach-load-balancer-target-groups`.

### AWS CLI

Per scollegare un gruppo target di sistemi di bilanciamento del carico da un gruppo Auto Scaling

Questo esempio scollega il gruppo target del sistema di bilanciamento del carico specificato dal gruppo Auto Scaling specificato.

```

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

```

Questo comando non produce alcun output

Per ulteriori informazioni, consulta [Collegare un sistema di bilanciamento del carico al gruppo Auto Scaling nella Amazon Auto Scaling User EC2 Guide](#).

- Per i API dettagli, consulta [DetachLoadBalancerTargetGroups](#) Command Reference. AWS CLI

## detach-load-balancers

Il seguente esempio di codice mostra come utilizzare `detach-load-balancers`.

## AWS CLI

Per scollegare un Classic Load Balancer da un gruppo Auto Scaling

Questo esempio scollega il Classic Load Balancer specificato dal gruppo Auto Scaling specificato.

```
aws autoscaling detach-load-balancers \  
  --load-balancer-names my-load-balancer \  
  --auto-scaling-group-name my-asg
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Collegare un sistema di bilanciamento del carico al gruppo Auto Scaling nella Amazon Auto Scaling User EC2 Guide](#).

- Per i API dettagli, consulta [DetachLoadBalancers](#) Command Reference.AWS CLI

## disable-metrics-collection

Il seguente esempio di codice mostra come utilizzare `disable-metrics-collection`.

### AWS CLI

Per disabilitare la raccolta di metriche per un gruppo di Auto Scaling

Questo esempio disabilita la raccolta della `GroupDesiredCapacity` metrica per il gruppo Auto Scaling specificato.

```
aws autoscaling disable-metrics-collection \  
  --auto-scaling-group-name my-asg \  
  --metrics GroupDesiredCapacity
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i [CloudWatch parametri di monitoraggio per i gruppi e le istanze di Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per i API dettagli, consulta [DisableMetricsCollection](#) Command Reference.AWS CLI

## enable-metrics-collection

Il seguente esempio di codice mostra come utilizzare `enable-metrics-collection`.

## AWS CLI

Esempio 1: abilitare la raccolta di metriche per un gruppo di Auto Scaling

Questo esempio abilita la raccolta di dati per il gruppo Auto Scaling specificato.

```
aws autoscaling enable-metrics-collection \  
  --auto-scaling-group-name my-asg \  
  --granularity "1Minute"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i [CloudWatch parametri di monitoraggio per i gruppi e le istanze di Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 2: raccogliere dati per la metrica specificata per un gruppo Auto Scaling

Per raccogliere dati per una metrica specifica, utilizza l'opzione. `--metrics`

```
aws autoscaling enable-metrics-collection \  
  --auto-scaling-group-name my-asg \  
  --metrics GroupDesiredCapacity --granularity "1Minute"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i [CloudWatch parametri di monitoraggio per i gruppi e le istanze di Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per i API dettagli, consulta [EnableMetricsCollection](#) Command Reference.AWS CLI

## enter-standby

Il seguente esempio di codice mostra come utilizzare `enter-standby`.

## AWS CLI

Per spostare le istanze in modalità standby

Questo esempio mette l'istanza specificata in modalità standby. Ciò è utile per aggiornare o risolvere i problemi di un'istanza attualmente in servizio.

```
aws autoscaling enter-standby \  
  --auto-scaling-group-name my-asg \  
  --instance-ids i-1234567890
```



```
--instance-ids i-061c63c5eb45f0416 \  
--auto-scaling-group-name my-asg \  
--should-decrement-desired-capacity
```

Output:

```
{  
  "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\"}"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta il [ciclo di vita delle istanze di Amazon EC2 Auto Scaling nella Amazon Auto Scaling EC2 User Guide](#).

- Per i API dettagli, consulta [EnterStandby](#) Command Reference.AWS CLI

## execute-policy

Il seguente esempio di codice mostra come utilizzare `execute-policy`.

AWS CLI

Per eseguire una politica di scalabilità

Questo esempio esegue il criterio di ridimensionamento denominato `my-step-scale-out-policy` per il gruppo Auto Scaling specificato.

```
aws autoscaling execute-policy \  
--auto-scaling-group-name my-asg \  
--policy-name my-step-scale-out-policy \  

```

```
--metric-value 95 \  
--breach-threshold 80
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Step and simple scaling policies](#) nella Amazon EC2 Auto Scaling User Guide.

- Per API i dettagli, consulta AWS CLI Command [ExecutePolicy](#) Reference.

## exit-standby

Il seguente esempio di codice mostra come utilizzare `exit-standby`.

### AWS CLI

Per spostare le istanze dalla modalità standby

Questo esempio sposta l'istanza specificata fuori dalla modalità standby.

```
aws autoscaling exit-standby \  
--instance-ids i-061c63c5eb45f0416 \  
--auto-scaling-group-name my-asg
```

Output:

```
{  
  "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\"}"  
    }  
  ]  
}
```

```
]
}
```

Per ulteriori informazioni, consulta [Rimuovere temporaneamente le istanze dal gruppo Auto Scaling nella Amazon Auto Scaling EC2](#) User Guide.

- Per API i dettagli, consulta Command [ExitStandby](#)Reference AWS CLI .

## put-lifecycle-hook

Il seguente esempio di codice mostra come utilizzare `put-lifecycle-hook`.

### AWS CLI

Esempio 1: Per creare un hook del ciclo di vita

Questo esempio crea un lifecycle hook che verrà richiamato su tutte le istanze appena avviate, con un timeout di 4800 secondi. Ciò è utile per mantenere le istanze in uno stato di attesa fino al termine degli script dei dati utente o per richiamare una funzione Lambda utilizzando AWS . 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
```

Questo comando non produce alcun output. Se esiste già un lifecycle hook con lo stesso nome, verrà sovrascritto dal nuovo lifecycle hook.

Per ulteriori informazioni, consulta i riferimenti al [ciclo di vita di Amazon EC2 Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 2: inviare un messaggio SNS e-mail Amazon per informarti delle transizioni di stato dell'istanza

Questo esempio crea un hook del ciclo di vita con l'SNSargomento e il IAM ruolo Amazon da utilizzare per ricevere notifiche all'avvio dell'istanza.

```
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
```

```
--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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i riferimenti al [ciclo di vita di Amazon EC2 Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 3: pubblicare un messaggio in una SQS coda Amazon

Questo esempio crea un lifecycle hook che pubblica un messaggio con metadati nella coda Amazon specificata. SQS

```
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"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i riferimenti al [ciclo di vita di Amazon EC2 Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per i API dettagli, consulta Command Reference. [PutLifecycleHookAWS CLI](#)

## put-notification-configuration

Il seguente esempio di codice mostra come utilizzare `put-notification-configuration`.

AWS CLI

Per aggiungere una notifica

Questo esempio aggiunge la notifica specificata al gruppo Auto Scaling specificato.

```
aws autoscaling put-notification-configuration \  
  --auto-scaling-group-name my-asg \  
  --notification-configuration my-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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Ricevere SNS notifiche Amazon quando il tuo gruppo Auto Scaling aumenta nella Amazon EC2 Auto Scaling User Guide](#).

- Per API i dettagli, consulta Command [PutNotificationConfiguration](#) Reference AWS CLI .

## put-scaling-policy

Il seguente esempio di codice mostra come utilizzare `put-scaling-policy`.

### AWS CLI

Per aggiungere una politica di ridimensionamento del tracciamento degli obiettivi a un gruppo Auto Scaling

L'esempio seguente applica una politica di ridimensionamento del tracciamento delle destinazioni al gruppo Auto Scaling specificato. L'output contiene i nomi ARNs e i nomi dei due CloudWatch allarmi creati per tuo conto. Se esiste già una politica di scalabilità con lo stesso nome, verrà sovrascritta dalla nuova politica di scalabilità.

```
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
```

Contenuto di `config.json`.

```
{  
  "TargetValue": 1000.0,  
  "PredefinedMetricSpecification": {  
    "PredefinedMetricType": "ALBRequestCountPerTarget",  
    "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-alb-target-group/943f017f100becff"  
  }  
}
```

**Output:**

```
{
  "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"
    }
  ]
}
```

Per altri esempi, consulta [Example scaling policies for the AWS Command Line Interface \(AWS CLI\)](#) nella Amazon EC2 Auto Scaling User Guide.

- Per API i dettagli, consulta AWS CLI Command [PutScalingPolicy](#) Reference.

**put-scheduled-update-group-action**

Il seguente esempio di codice mostra come utilizzare `put-scheduled-update-group-action`.

**AWS CLI**

Esempio 1: aggiungere un'azione pianificata a un gruppo Auto Scaling

Questo esempio aggiunge l'azione pianificata specificata al gruppo Auto Scaling specificato.

```
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
```

Questo comando non produce alcun output. Se esiste già un'azione pianificata con lo stesso nome, verrà sovrascritta dalla nuova azione pianificata.

Per altri esempi, consulta [Scheduled Scaling](#) nella Amazon EC2 Auto Scaling User Guide.

Esempio 2: Per specificare una pianificazione ricorrente

Questo esempio crea un'azione pianificata da scalare in base a una pianificazione ricorrente la cui esecuzione è programmata alle 00:30 del primo gennaio, giugno e dicembre di ogni anno.

```
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
```

Questo comando non produce alcun output. Se esiste già un'azione pianificata con lo stesso nome, verrà sovrascritta dalla nuova azione pianificata.

Per altri esempi, consulta [Scheduled Scaling](#) nella Amazon EC2 Auto Scaling User Guide.

- Per API i dettagli, consulta AWS CLI Command [PutScheduledUpdateGroupActionReference](#).

## put-warm-pool

Il seguente esempio di codice mostra come utilizzare `put-warm-pool`.

### AWS CLI

Per creare una piscina calda

L'esempio seguente crea un pool caldo per il gruppo Auto Scaling specificato.

```
aws autoscaling put-warm-pool \  
  --auto-scaling-group-name my-asg \  
  --min-size 2
```

Questo comando non produce alcun output. Se esiste già un pool caldo, verrà aggiornato.

Per ulteriori informazioni, consulta [Warm pools for Amazon EC2 Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per API i dettagli, consulta AWS CLI Command [PutWarmPoolReference](#).

## **record-lifecycle-action-heartbeat**

Il seguente esempio di codice mostra come utilizzare `record-lifecycle-action-heartbeat`.

### AWS CLI

Per registrare un'azione del ciclo di vita (heartbeat)

Questo esempio registra un heartbeat di un'azione del ciclo di vita per mantenere l'istanza in sospenso.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i riferimenti al [ciclo di vita di Amazon EC2 Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

- Per i API dettagli, consulta Command Reference. [RecordLifecycleActionHeartbeat](#) AWS CLI

## **resume-processes**

Il seguente esempio di codice mostra come utilizzare `resume-processes`.

### AWS CLI

Per riprendere i processi sospesi

Questo esempio riprende il processo di ridimensionamento sospeso specificato per il gruppo Auto Scaling specificato.

```
aws autoscaling resume-processes \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
```



```
--scaling-processes AlarmNotification
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Sospensione e ripresa dei processi di scalabilità nella Amazon Auto Scaling](#) User Guide. EC2

- Per i API dettagli, consulta Command Reference. [ResumeProcesses](#)AWS CLI

## rollback-instance-refresh

Il seguente esempio di codice mostra come utilizzare `rollback-instance-refresh`.

AWS CLI

Per ripristinare l'aggiornamento di un'istanza

L'`rollback-instance-refresh` esempio seguente ripristina un aggiornamento dell'istanza in corso per il gruppo Auto Scaling specificato.

```
aws autoscaling rollback-instance-refresh \  
  --auto-scaling-group-name my-asg
```

Output:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

Per ulteriori informazioni, consulta [Annulla le modifiche con un rollback](#) nella Amazon Auto EC2 Scaling User Guide.

- Per API i dettagli, consulta Command [RollbackInstanceRefresh](#)Reference AWS CLI .

## set-desired-capacity

Il seguente esempio di codice mostra come utilizzare `set-desired-capacity`.

AWS CLI

Per impostare la capacità desiderata per un gruppo Auto Scaling

Questo esempio imposta la capacità desiderata per il gruppo Auto Scaling specificato.

```
aws autoscaling set-desired-capacity \  
  --auto-scaling-group-name my-asg \  
  --desired-capacity 2 \  
  --honor-cooldown
```

In caso di esito positivo, il comando torna al prompt.

- Per API i dettagli, vedere [SetDesiredCapacity](#) in AWS CLI Command Reference.

## set-instance-health

Il seguente esempio di codice mostra come utilizzare `set-instance-health`.

### AWS CLI

Per impostare lo stato di integrità di un'istanza

Questo esempio imposta lo stato di integrità dell'istanza specificata su `Unhealthy`.

```
aws autoscaling set-instance-health \  
  --instance-id i-061c63c5eb45f0416 \  
  --health-status Unhealthy
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [SetInstanceHealth](#) in AWS CLI Command Reference.

## set-instance-protection

Il seguente esempio di codice mostra come utilizzare `set-instance-protection`.

### AWS CLI

Esempio 1: per abilitare l'impostazione di protezione dell'istanza per un'istanza

Questo esempio abilita la protezione dell'istanza per l'istanza specificata.

```
aws autoscaling set-instance-protection \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg --protected-from-scale-in
```

Questo comando non produce alcun output.

Esempio 2: per disabilitare l'impostazione di protezione dell'istanza per un'istanza

Questo esempio disattiva la protezione dell'istanza per l'istanza specificata.

```
aws autoscaling set-instance-protection \  
  --instance-ids i-061c63c5eb45f0416 \  
  --auto-scaling-group-name my-asg \  
  --no-protected-from-scale-in
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [SetInstanceProtection](#) in AWS CLI Command Reference.

## start-instance-refresh

Il seguente esempio di codice mostra come utilizzare `start-instance-refresh`.

AWS CLI

Esempio 1: per avviare l'aggiornamento di un'istanza utilizzando i parametri della riga di comando

L'`start-instance-refresh` seguente avvia l'aggiornamento di un'istanza utilizzando argomenti della riga di comando. Il `preferences` parametro opzionale specifica un numero `InstanceWarmup` di 60 secondi e uno `MinHealthyPercentage` di 50 percentuale.

```
aws autoscaling start-instance-refresh \  
  --auto-scaling-group-name my-asg \  
  --preferences '{"InstanceWarmup": 60, "MinHealthyPercentage": 50}'
```

Output:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

Per ulteriori informazioni, consulta [Start an instance refresh](#) nella Amazon EC2 Auto Scaling User Guide.

Esempio 2: per avviare l'aggiornamento di un'istanza utilizzando un file JSON

L'`start-instance-refresh` seguente avvia l'aggiornamento di un'istanza utilizzando un JSON file. È possibile specificare il gruppo Auto Scaling e definire la configurazione e le preferenze desiderate in un JSON file, come illustrato nell'esempio seguente.

```
aws autoscaling start-instance-refresh \  
  --cli-input-json file://config.json
```

Contenuto di `config.json`.

```
{  
  "AutoScalingGroupName": "my-asg",  
  "DesiredConfiguration": {  
    "LaunchTemplate": {  
      "LaunchTemplateId": "lt-068f72b729example",  
      "Version": "$Default"  
    }  
  },  
  "Preferences": {  
    "InstanceWarmup": 60,  
    "MinHealthyPercentage": 50,  
    "AutoRollback": true,  
    "ScaleInProtectedInstances": Ignore,  
    "StandbyInstances": Terminate  
  }  
}
```

Output:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

Per ulteriori informazioni, consulta [Start an instance refresh](#) nella Amazon EC2 Auto Scaling User Guide.

- Per API i dettagli, consulta AWS CLI Command [StartInstanceRefresh](#) Reference.

## **suspend-processes**

Il seguente esempio di codice mostra come utilizzare `suspend-processes`.

## AWS CLI

Per sospendere i processi di Auto Scaling

Questo esempio sospende il processo di ridimensionamento specificato per il gruppo Auto Scaling specificato.

```
aws autoscaling suspend-processes \  
  --auto-scaling-group-name my-asg \  
  --scaling-processes AlarmNotification
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Sospensione e ripresa dei processi di scalabilità nella Amazon Auto Scaling](#) User Guide. EC2

- Per i API dettagli, consulta Command Reference. [SuspendProcesses](#) AWS CLI

## terminate-instance-in-auto-scaling-group

Il seguente esempio di codice mostra come utilizzare `terminate-instance-in-auto-scaling-group`.

## AWS CLI

Per terminare un'istanza in un gruppo Auto Scaling

Questo esempio termina l'istanza specificata dal gruppo Auto Scaling specificato senza aggiornare la dimensione del gruppo. Amazon EC2 Auto Scaling avvia un'istanza sostitutiva al termine dell'istanza specificata.

```
aws autoscaling terminate-instance-in-auto-scaling-group \  
  --instance-id i-061c63c5eb45f0416 \  
  --no-should-decrement-desired-capacity
```

Output:

```
{  
  "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\"}"
  }
]
}

```

- Per API i dettagli, consulta Command [TerminateInstanceInAutoScalingGroup](#) Reference AWS CLI .

## update-auto-scaling-group

Il seguente esempio di codice mostra come utilizzare `update-auto-scaling-group`.

### AWS CLI

Esempio 1: Per aggiornare i limiti di dimensione di un gruppo di Auto Scaling

Questo esempio aggiorna il gruppo Auto Scaling specificato con una dimensione minima di 2 e una dimensione massima di 10.

```

aws autoscaling update-auto-scaling-group \
  --auto-scaling-group-name my-asg \
  --min-size 2 \
  --max-size 10

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Impostazione dei limiti di capacità per il tuo gruppo Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 2: aggiungere i controlli di integrità di Elastic Load Balancing e specificare quali zone di disponibilità e sottoreti utilizzare

Questo esempio aggiorna il gruppo Auto Scaling specificato per aggiungere i controlli di integrità Elastic Load Balancing. Questo comando aggiorna anche il valore di `--vpc-zone-identifier` con un elenco di sottoreti IDs in più zone di disponibilità.

```
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"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Elastic Load Balancing e Amazon Auto EC2 Scaling nella Amazon Auto EC2 Scaling User Guide](#).

Esempio 3: per aggiornare il gruppo di collocamento e la politica di cessazione

Questo esempio aggiorna il gruppo di collocamento e la politica di licenziamento da utilizzare.

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --placement-group my-placement-group \  
  --termination-policies "OldestInstance"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i [gruppi Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Esempio 4: per utilizzare la versione più recente del modello di lancio

Questo esempio aggiorna il gruppo Auto Scaling specificato per utilizzare la versione più recente del modello di avvio specificato.

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Launch templates](#) nella Amazon EC2 Auto Scaling User Guide.

Esempio 5: per utilizzare una versione specifica del modello di lancio

Questo esempio aggiorna il gruppo Auto Scaling specificato per utilizzare una versione specifica di un modello di avvio anziché la versione più recente o predefinita.

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='2'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Launch templates](#) nella Amazon EC2 Auto Scaling User Guide.

Esempio 6: definire una politica a istanze miste e abilitare il ribilanciamento della capacità

Questo esempio aggiorna il gruppo Auto Scaling specificato per utilizzare una politica di istanze miste e consente il ribilanciamento della capacità. Questa struttura consente di specificare gruppi con capacità Spot e On-Demand e utilizzare modelli di avvio diversi per architetture diverse.

```
aws autoscaling update-auto-scaling-group \  
  --cli-input-json file://~/config.json
```

Contenuto di 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"
  }
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i [gruppi di Auto Scaling con più tipi di istanze e opzioni di acquisto](#) nella Amazon Auto EC2 Scaling User Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateAutoScalingGroupReference](#).

## Esempi di Piani di Auto Scaling che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando i piani AWS Command Line Interface con Auto Scaling.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto degli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-scaling-plan**

Il seguente esempio di codice mostra come utilizzare `create-scaling-plan`.

AWS CLI

Per creare un piano di scalabilità

L'create-scaling-planesempio seguente crea un piano di scalabilità denominato my-scaling-plan utilizzando un JSON file già creato (denominato config.json). La struttura del piano di scalabilità include un'istruzione di ridimensionamento per un gruppo Auto Scaling denominato my-asg. Specifica la proprietà TagFilters come origine dell'applicazione e abilita il dimensionamento predittivo e il dimensionamento dinamico.

```
aws autoscaling-plans create-scaling-plan \  
  --scaling-plan-name my-scaling-plan \  
  --cli-input-json file://~/config.json
```

Contenuto del file: config.json

```
{  
  "ApplicationSource": {  
    "TagFilters": [  
      {  
        "Key": "purpose",  
        "Values": [  
          "my-application"  
        ]  
      }  
    ]  
  },  
  "ScalingInstructions": [  
    {  
      "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
  }
]
}
```

Output:

```
{
  "ScalingPlanVersion": 1
}
```

Per ulteriori informazioni, consulta la Guida per l'[utente di AWS Auto Scaling](#).

- Per API i dettagli, vedere [CreateScalingPlan](#) in AWS CLI Command Reference.

## delete-scaling-plan

Il seguente esempio di codice mostra come utilizzare `delete-scaling-plan`.

AWS CLI

Per eliminare un piano di scalabilità

L'`delete-scaling-plan` seguente elimina il piano di scala specificato.

```
aws autoscaling-plans delete-scaling-plan \
  --scaling-plan-name my-scaling-plan \
  --scaling-plan-version 1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la Guida per l'[utente di AWS Auto Scaling](#).

- Per API i dettagli, vedere [DeleteScalingPlan](#) in AWS CLI Command Reference.

## describe-scaling-plan-resources

Il seguente esempio di codice mostra come utilizzare `describe-scaling-plan-resources`.

## AWS CLI

Per descrivere le risorse scalabili per un piano di scalabilità

L'output di `aws autoscaling-plans describe-scaling-plan-resources` seguente visualizza i dettagli sulla singola risorsa scalabile (un gruppo Auto Scaling) associata al piano di scalabilità specificato.

```
aws autoscaling-plans describe-scaling-plan-resources \  
  --scaling-plan-name my-scaling-plan \  
  --scaling-plan-version 1
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Che cos'è l' AWS Auto Scaling?](#) nella Guida per l'utente di AWS Auto Scaling.

- Per API i dettagli, vedere [DescribeScalingPlanResources](#) in AWS CLI Command Reference.

## describe-scaling-plans

Il seguente esempio di codice mostra come utilizzare `describe-scaling-plans`.

### AWS CLI

Per descrivere un piano di scalabilità

L'`describe-scaling-plans` esempio seguente visualizza i dettagli del piano di scalabilità specificato.

```
aws autoscaling-plans describe-scaling-plans \  
  --scaling-plan-names scaling-plan-with-asg-and-ddb
```

Output:

```
{  
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Che cos'è l' AWS Auto Scaling?](#) nella Guida per l'utente di AWS Auto Scaling.

- Per API i dettagli, vedere [DescribeScalingPlans](#) in AWS CLI Command Reference.

## get-scaling-plan-resource-forecast-data

Il seguente esempio di codice mostra come utilizzare `get-scaling-plan-resource-forecast-data`.

### AWS CLI

Per recuperare i dati di previsione del carico

Questo esempio recupera i dati di previsione del carico per una risorsa scalabile (un gruppo Auto Scaling) associata al piano di scalabilità specificato.

```

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"

```

Output:

```

{
  "Datapoints": [...]
}

```

Per ulteriori informazioni, consulta [Cos'è l' AWS Auto Scaling](#) nella Guida per l'utente AWS di Auto Scaling.

- Per API i dettagli, vedere [GetScalingPlanResourceForecastData](#) in AWS CLI Command Reference.

## update-scaling-plan

Il seguente esempio di codice mostra come utilizzare `update-scaling-plan`.

AWS CLI

Per aggiornare un piano di scalabilità

L'update-scaling-plan esempio seguente modifica la metrica di scala per un gruppo Auto Scaling nel piano di scala specificato.

```

aws autoscaling-plans update-scaling-plan \
  --scaling-plan-name my-scaling-plan \
  --scaling-plan-version 1 \
  --scaling-instructions
  '{"ScalableDimension":"autoscaling:autoScalingGroup:DesiredCapacity","ResourceId":"autoScalingGroup/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}'

```



Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Che cos'è l' AWS Auto Scaling?](#) nella Guida per l'utente di AWS Auto Scaling.

- Per API i dettagli, vedere [UpdateScalingPlan](#) in AWS CLI Command Reference.

## AWS Backup esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Backup.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, dove puoi trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-backup-plan**

Il seguente esempio di codice mostra come utilizzare `create-backup-plan`.

AWS CLI

Per creare un piano di backup

L'`create-backup-plan` esempio seguente crea il piano di backup specificato con una conservazione di 35 giorni.

```
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}}]}"
```

## Output:

```
{
  "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"
}
```

Per ulteriori informazioni, vedere [Creazione di un piano di backup](#) nella AWS Backup Developer Guide.

- Per API i dettagli, vedere [CreateBackupPlan](#) in AWS CLI Command Reference.

## create-backup-vault

Il seguente esempio di codice mostra come utilizzare `create-backup-vault`.

### AWS CLI

Per creare un archivio di backup

L'`create-backup-vault` esempio seguente crea un archivio di backup con il nome specificato.

```
aws backup create-backup-vault
--backup-vault-name sample-vault
```

Questo comando non produce alcun output. Output:

```
{
  "BackupVaultName": "sample-vault",
  "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-vault:sample-
vault",
  "CreationDate": 1568928338.385
}
```

Per ulteriori informazioni, vedere [Creating a Backup Vault](#) nella AWS Backup Developer Guide.

- Per API i dettagli, vedere [CreateBackupVault](#) in AWS CLI Command Reference.

## get-backup-plan-from-template

Il seguente esempio di codice mostra come utilizzare `get-backup-plan-from-template`.

### AWS CLI

Per ottenere un piano di backup esistente da un modello

L'esempio seguente ottiene un piano di backup esistente da un modello che specifica un backup giornaliero con una conservazione di 35 giorni.

```
aws backup get-backup-plan-from-template \
  --backup-plan-template-id "87c0c1ef-254d-4180-8fef-2e76a2c38aaa"
```

Output:

```
{
  "BackupPlanDocument": {
    "Rules": [
      {
        "RuleName": "DailyBackups",
        "ScheduleExpression": "cron(0 5 ? * * *)",
        "StartWindowMinutes": 480,
        "Lifecycle": {
          "DeleteAfterDays": 35
        }
      }
    ]
  }
}
```

Per ulteriori informazioni, vedere [Creazione di un piano di backup](#) nella AWS Backup Developer Guide.

- Per API i dettagli, vedere [GetBackupPlanFromTemplate](#) in AWS CLI Command Reference.

## get-backup-plan

Il seguente esempio di codice mostra come utilizzare `get-backup-plan`.

### AWS CLI

Per ottenere i dettagli di un piano di backup

L'get-backup-planesempio seguente visualizza i dettagli del piano di backup specificato.

```
aws backup get-backup-plan \  
--backup-plan-id "fcbf5d8f-bd77-4f3a-9c97-f24fb3d373a5"
```

Output:

```
{  
  "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  
}
```

Per ulteriori informazioni, vedere [Creazione di un piano di backup](#) nella AWS Backup Developer Guide.

- Per API i dettagli, vedere [GetBackupPlan](#) in AWS CLI Command Reference.

## list-backup-jobs

Il seguente esempio di codice mostra come utilizzare `list-backup-jobs`.

### AWS CLI

Esempio 1: per elencare tutti i processi di backup

L'`list-backup-jobs` seguente restituisce i metadati relativi ai processi di backup nel tuo AWS account.

```
aws backup list-backup-jobs
```

Output:

```
{
  "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"
    }
  ]
}
```

```
]
}
```

Per ulteriori informazioni, vedere [Creating a Backup nella AWS Backup Developer Guide](#).

Esempio 2: per elencare i processi di backup completati

L'`list-backup-jobs` seguente restituisce i metadati relativi ai processi di backup completati nel tuo AWS account.

```
aws backup list-backup-jobs \
  --by-state COMPLETED
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Creating a Backup nella AWS Backup Developer Guide](#).

- Per API i dettagli, vedere [ListBackupJobs](#) in AWS CLI Command Reference.

## AWS Batch esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Batch.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **cancel-job**

Il seguente esempio di codice mostra come utilizzare `cancel-job`.

AWS CLI

Per annullare un lavoro

Questo esempio annulla un lavoro con l'ID del lavoro specificato.

Comando:

```
aws batch cancel-job --job-id bcf0b186-a532-4122-842e-2ccab8d54efb --  
reason "Cancelling job."
```

- Per API i dettagli, vedere [CancelJob](#) in AWS CLI Command Reference.

### **create-compute-environment**

Il seguente esempio di codice mostra come utilizzare `create-compute-environment`.

AWS CLI

Per creare un ambiente di elaborazione gestito con istanze On-Demand

Questo esempio crea un ambiente di elaborazione gestito con tipi di istanze C4 specifici che vengono lanciati su richiesta. L'ambiente di calcolo si chiama C4. OnDemand

Comando:

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/C4OnDemand.json
```

JSONformato di file:

```
{
  "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"
}
```



## Output:

```
{
  "computeEnvironmentName": "C4OnDemand",
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/C4OnDemand"
}
```

Per creare un ambiente di elaborazione gestito con istanze Spot

Questo esempio crea un ambiente di elaborazione gestito con il tipo di istanza M4 che viene lanciato quando il prezzo dell'offerta Spot è pari o inferiore al 20% del prezzo On-Demand per il tipo di istanza. L'ambiente di calcolo si chiama M4Spot.

Comando:

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/
M4Spot.json
```

JSONformato di file:

```
{
  "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"
}
```

Output:

```
{
  "computeEnvironmentName": "M4Spot",
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/M4Spot"
}
```

- Per API i dettagli, vedere [CreateComputeEnvironment](#) in AWS CLI Command Reference.

## create-job-queue

Il seguente esempio di codice mostra come utilizzare `create-job-queue`.

### AWS CLI

Per creare una coda di lavoro a bassa priorità con un unico ambiente di elaborazione

Questo esempio crea una coda di lavoro chiamata `LowPriority` che utilizza l'ambiente di calcolo `M4Spot`.

Comando:

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/  
LowPriority.json
```

JSONformato di file:

```
{
  "jobQueueName": "LowPriority",
  "state": "ENABLED",
  "priority": 10,
```

```
"computeEnvironmentOrder": [  
  {  
    "order": 1,  
    "computeEnvironment": "M4Spot"  
  }  
]  
}
```

Output:

```
{  
  "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/LowPriority",  
  "jobQueueName": "LowPriority"  
}
```

Per creare una coda di lavoro ad alta priorità con due ambienti di elaborazione

Questo esempio crea una coda di lavoro chiamata HighPriority che utilizza l'ambiente di OnDemand calcolo C4 con un ordine di 1 e l'ambiente di calcolo M4Spot con un ordine di 2. Lo scheduler tenterà innanzitutto di inserire i lavori nell'ambiente di calcolo C4. OnDemand

Comando:

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/  
HighPriority.json
```

JSONformato di file:

```
{  
  "jobQueueName": "HighPriority",  
  "state": "ENABLED",  
  "priority": 1,  
  "computeEnvironmentOrder": [  
    {  
      "order": 1,  
      "computeEnvironment": "C4OnDemand"  
    },  
    {  
      "order": 2,  
      "computeEnvironment": "M4Spot"  
    }  
  ]  
}
```

```
}
```

Output:

```
{
  "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/HighPriority",
  "jobQueueName": "HighPriority"
}
```

- Per API i dettagli, vedere [CreateJobQueue](#) in AWS CLI Command Reference.

## delete-compute-environment

Il seguente esempio di codice mostra come utilizzare `delete-compute-environment`.

AWS CLI

Per eliminare un ambiente di calcolo

Questo esempio elimina l'ambiente di calcolo `P2OnDemand`.

Comando:

```
aws batch delete-compute-environment --compute-environment P2OnDemand
```

- Per API i dettagli, vedere [DeleteComputeEnvironment](#) in AWS CLI Command Reference.

## delete-job-queue

Il seguente esempio di codice mostra come utilizzare `delete-job-queue`.

AWS CLI

Per eliminare una coda di lavori

Questo esempio elimina la coda dei GPGPU lavori.

Comando:

```
aws batch delete-job-queue --job-queue GPGPU
```

- Per API i dettagli, vedere [DeleteJobQueue](#) in AWS CLI Command Reference.

## deregister-job-definition

Il seguente esempio di codice mostra come utilizzare `deregister-job-definition`.

### AWS CLI

Per annullare la registrazione di una definizione di processo

Questo esempio annulla la registrazione di una definizione di lavoro denominata `sleep10`.

Comando:

```
aws batch deregister-job-definition --job-definition sleep10
```

- Per i API dettagli, vedere [DeregisterJobDefinition](#) in Command Reference.AWS CLI

## describe-compute-environments

Il seguente esempio di codice mostra come utilizzare `describe-compute-environments`.

### AWS CLI

Per descrivere un ambiente di calcolo

Questo esempio descrive l'ambiente di OnDemand calcolo P2.

Comando:

```
aws batch describe-compute-environments --compute-environments P2OnDemand
```

Output:

```
{
  "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/
P2OnDemand_Batch_2c06f29d-d1fe-3a49-879d-42394c86effc",
    "state": "ENABLED",
    "computeEnvironmentName": "P2OnDemand",
    "type": "MANAGED"
    }
    ]
}

```

- Per API i dettagli, vedere [DescribeComputeEnvironments](#) in AWS CLI Command Reference.

## describe-job-definitions

Il seguente esempio di codice mostra come utilizzare `describe-job-definitions`.

### AWS CLI

Per descrivere le definizioni delle mansioni attive

Questo esempio descrive tutte le definizioni delle mansioni attive.

Comando:

```
aws batch describe-job-definitions --status ACTIVE
```

Output:

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeJobDefinitions](#) in AWS CLI Command Reference.

## describe-job-queues

Il seguente esempio di codice mostra come utilizzare `describe-job-queues`.

AWS CLI

Per descrivere una coda di lavoro

Questo esempio descrive la coda dei HighPriority lavori.

Comando:

```
aws batch describe-job-queues --job-queues HighPriority
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeJobQueues](#) in AWS CLI Command Reference.

## describe-jobs

Il seguente esempio di codice mostra come utilizzare `describe-jobs`.

AWS CLI

Descrivere un lavoro

L'`describe-jobs`esempio seguente descrive un lavoro con l'ID del lavoro specificato.

```
aws batch describe-jobs \  
  --jobs bcf0b186-a532-4122-842e-2ccab8d54efb
```



**Output:**

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeJobs](#) in AWS CLI Command Reference.

**list-jobs**

Il seguente esempio di codice mostra come utilizzare `list-jobs`.

**AWS CLI**

Per elencare i lavori in esecuzione

Questo esempio elenca i processi in esecuzione nella coda dei HighPriority processi.

Comando:

```
aws batch list-jobs --job-queue HighPriority
```

Output:

```
{
  "jobSummaryList": [
    {
      "jobName": "example",
      "jobId": "e66ff5fd-a1ff-4640-b1a2-0b0a142f49bb"
    }
  ]
}
```

Per elencare i lavori inviati

Questo esempio elenca i lavori nella coda dei HighPriority lavori che si trovano nello stato del SUBMITTED lavoro.

Comando:

```
aws batch list-jobs --job-queue HighPriority --job-status SUBMITTED
```

Output:

```
{
  "jobSummaryList": [
    {
      "jobName": "example",
      "jobId": "68f0c163-fbd4-44e6-9fd1-25b14a434786"
    }
  ]
}
```

- Per API i dettagli, vedere [ListJobs](#) in AWS CLI Command Reference.

## register-job-definition

Il seguente esempio di codice mostra come utilizzare `register-job-definition`.

## AWS CLI

Per registrare una definizione di lavoro

Questo esempio registra una definizione di processo per un semplice processo contenitore.

Comando:

```
aws batch register-job-definition --job-definition-name sleep30 --type container --
container-properties '{ "image": "busybox", "vcpus": 1, "memory": 128, "command":
[ "sleep", "30"]}'
```

Output:

```
{
  "jobDefinitionArn": "arn:aws:batch:us-east-1:012345678910:job-definition/
sleep30:1",
  "jobDefinitionName": "sleep30",
  "revision": 1
}
```

- Per API i dettagli, vedere [RegisterJobDefinition](#) in AWS CLI Command Reference.

## submit-job

Il seguente esempio di codice mostra come utilizzare `submit-job`.

## AWS CLI

Per inviare un lavoro

Questo esempio invia un semplice processo contenitore chiamato `example` alla coda dei `HighPriority` lavori.

Comando:

```
aws batch submit-job --job-name example --job-queue HighPriority --job-
definition sleep60
```

Output:

```
{
```

```
"jobName": "example",  
"jobId": "876da822-4198-45f2-a252-6cea32512ea8"  
}
```

- Per API i dettagli, vedere [SubmitJob](#) in AWS CLI Command Reference.

## terminate-job

Il seguente esempio di codice mostra come utilizzare `terminate-job`.

### AWS CLI

Per terminare un lavoro

Questo esempio termina un lavoro con l'ID di lavoro specificato.

Comando:

```
aws batch terminate-job --job-id 61e743ed-35e4-48da-b2de-5c8333821c84 --  
reason "Terminating job."
```

- Per API i dettagli, vedere [TerminateJob](#) in AWS CLI Command Reference.

## update-compute-environment

Il seguente esempio di codice mostra come utilizzare `update-compute-environment`.

### AWS CLI

Per aggiornare un ambiente di calcolo

Questo esempio disabilita l'ambiente di OnDemand calcolo P2 in modo che possa essere eliminato.

Comando:

```
aws batch update-compute-environment --compute-environment P2OnDemand --  
state DISABLED
```

Output:

```
{
```

```
"computeEnvironmentName": "P2OnDemand",
"computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/P2OnDemand"
}
```

- Per API i dettagli, vedere [UpdateComputeEnvironment](#) in AWS CLI Command Reference.

## update-job-queue

Il seguente esempio di codice mostra come utilizzare `update-job-queue`.

### AWS CLI

Per aggiornare una coda di lavori

Questo esempio disabilita una coda di lavori in modo che possa essere eliminata.

Comando:

```
aws batch update-job-queue --job-queue GPGPU --state DISABLED
```

Output:

```
{
  "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/GPGPU",
  "jobQueueName": "GPGPU"
}
```

- Per API i dettagli, vedere [UpdateJobQueue](#) in AWS CLI Command Reference.

## Budget AWS esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Budget AWS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### create-budget

Il seguente esempio di codice mostra come utilizzare `create-budget`.

#### AWS CLI

Per creare un budget di costi e utilizzo

Il `create-budget` comando seguente crea un budget di costi e utilizzo.

```
aws budgets create-budget \  
  --account-id 111122223333 \  
  --budget file://budget.json \  
  --notifications-with-subscribers file://notifications-with-subscribers.json
```

Contenuto di `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"  
}
```

Contenuto di `notifications-with-subscribers.json`.

```
[  
  {  
    "Notification": {  
      "ComparisonOperator": "GREATER_THAN",  
      "NotificationType": "ACTUAL",  
      "Threshold": 80,  
      "ThresholdType": "PERCENTAGE"  
    },  
    "Subscribers": [  
      {  
        "Address": "example@example.com",  
        "SubscriptionType": "EMAIL"  
      }  
    ]  
  }  
]
```

- Per API i dettagli, vedere [CreateBudget](#) in AWS CLI Command Reference.

## **create-notification**

Il seguente esempio di codice mostra come utilizzare `create-notification`.

### AWS CLI

Per creare una notifica per il budget di costo e utilizzo specificato

Questo esempio crea una notifica per il budget di costi e utilizzo specificato.

Comando:

```
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
```

- Per API i dettagli, vedere [CreateNotification](#) in AWS CLI Command Reference.

## create-subscriber

Il seguente esempio di codice mostra come utilizzare `create-subscriber`.

AWS CLI

Per creare un sottoscrittore per una notifica associata a un budget di costo e utilizzo

Questo esempio crea un sottoscrittore per la notifica specificata.

Comando:

```
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
```

- Per API i dettagli, vedere [CreateSubscriber](#) in AWS CLI Command Reference.

## delete-budget

Il seguente esempio di codice mostra come utilizzare `delete-budget`.

AWS CLI

Per eliminare un budget di costi e utilizzo

Questo esempio elimina il budget di costi e utilizzo specificato.

Comando:

```
aws budgets delete-budget --account-id 111122223333 --budget-name "Example Budget"
```



- Per API i dettagli, vedere [DeleteBudget](#) in AWS CLI Command Reference.

## delete-notification

Il seguente esempio di codice mostra come utilizzare `delete-notification`.

### AWS CLI

Per eliminare una notifica da un budget

Questo esempio elimina la notifica specificata dal budget specificato.

Comando:

```
aws budgets delete-notification --account-id 111122223333 --budget-name "Example Budget" --notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENTAGE
```

- Per API i dettagli, vedere [DeleteNotification](#) in AWS CLI Command Reference.

## delete-subscriber

Il seguente esempio di codice mostra come utilizzare `delete-subscriber`.

### AWS CLI

Per eliminare un sottoscrittore da una notifica

Questo esempio elimina il sottoscrittore specificato dalla notifica specificata.

Comando:

```
aws budgets delete-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
```

- Per API i dettagli, vedere [DeleteSubscriber](#) in AWS CLI Command Reference.

## describe-budget

Il seguente esempio di codice mostra come utilizzare `describe-budget`.

## AWS CLI

Per recuperare un budget associato a un account

Questo esempio recupera il budget di costi e utilizzo specificato.

Comando:

```
aws budgets describe-budget --account-id 111122223333 --budget-name "Example Budget"
```

Output:

```
{
  "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"
      ]
    }
  }
}

```

- Per API i dettagli, vedere [DescribeBudget](#) in AWS CLI Command Reference.

## describe-budgets

Il seguente esempio di codice mostra come utilizzare `describe-budgets`.

### AWS CLI

Per recuperare i budget associati a un account

Questo esempio recupera i budget relativi ai costi e all'utilizzo di un account.

Comando:

```
aws budgets describe-budgets --account-id 111122223333 --max-results 20
```

Output:

```

{
  "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"
      ]
    }
  }
]
}

```

- Per API i dettagli, vedere [DescribeBudgets](#) in AWS CLI Command Reference.

## **describe-notifications-for-budget**

Il seguente esempio di codice mostra come utilizzare `describe-notifications-for-budget`.

### AWS CLI

Per recuperare le notifiche relative a un budget

Questo esempio recupera le notifiche relative a un budget di costi e utilizzo.

Comando:

```
aws budgets describe-notifications-for-budget --account-id 111122223333 --budget-name "Example Budget" --max-results 5
```

Output:

```
{
  "Notifications": [
    {
      "Threshold": 80.0,
      "ComparisonOperator": "GREATER_THAN",
      "NotificationType": "ACTUAL"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeNotificationsForBudget](#) in AWS CLI Command Reference.

## describe-subscribers-for-notification

Il seguente esempio di codice mostra come utilizzare `describe-subscribers-for-notification`.

AWS CLI

Per richiamare gli abbonati per una notifica sul budget

Questo esempio recupera gli abbonati per una notifica sul budget di costi e utilizzo.

Comando:

```
aws budgets describe-subscribers-for-notification --
account-id 111122223333 --budget-name "Example Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
--max-results 5
```

Output:

```
{
  "Subscribers": [
    {
      "SubscriptionType": "EMAIL",
      "Address": "example2@example.com"
    },
    {
      "SubscriptionType": "EMAIL",
      "Address": "example@example.com"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeSubscribersForNotification](#) in AWS CLI Command Reference.

## update-budget

Il seguente esempio di codice mostra come utilizzare `update-budget`.

### AWS CLI

Per sostituire un budget con un budget di costi e utilizzo

Questo esempio sostituisce un budget di costi e utilizzo con un nuovo budget.

Comando:

```
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"
}

```

- Per i API dettagli, vedere in Command Reference. [UpdateBudget](#) AWS CLI

## update-notification

Il seguente esempio di codice mostra come utilizzare `update-notification`.

### AWS CLI

Per sostituire una notifica per un budget di costi e utilizzo

Questo esempio sostituisce una notifica dell'80% per un budget di costi e utilizzo con una notifica del 90%.

Comando:

```

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

```

- Per API i dettagli, vedere [UpdateNotification](#) in AWS CLI Command Reference.

## update-subscriber

Il seguente esempio di codice mostra come utilizzare `update-subscriber`.

### AWS CLI

Per sostituire un abbonato per un budget relativo ai costi e all'utilizzo

Questo esempio sostituisce il sottoscrittore per un budget di costi e utilizzo.

Comando:

```
aws budgets update-subscriber --account-id 111122223333 --budget-name "Example Budget" --notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdType=PERCENTAGE --old-subscriber SubscriptionType=EMAIL,Address=example@example.com --new-subscriber SubscriptionType=EMAIL,Address=example2@example.com
```

- Per API i dettagli, vedere [UpdateSubscriber](#) in AWS CLI Command Reference.

## Esempi di utilizzo di Amazon Chime AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Chime. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### associate-phone-number-with-user

Il seguente esempio di codice mostra come utilizzare `associate-phone-number-with-user`.



## AWS CLI

Per associare un numero di telefono a un utente

L'associate-phone-number-with-user esempio seguente associa il numero di telefono specificato a un utente.

```
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"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Gestione dei numeri di telefono degli utenti](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [AssociatePhoneNumberWithUser AWS CLI Command Reference](#).

## associate-phone-numbers-with-voice-connector-group

Il seguente esempio di codice mostra come utilizzare associate-phone-numbers-with-voice-connector-group.

### AWS CLI

Per associare numeri di telefono a un gruppo Amazon Chime Voice Connector

L'associate-phone-numbers-with-voice-connector-group esempio seguente associa i numeri di telefono specificati a un gruppo Amazon Chime Voice Connector.

```
aws chime associate-phone-numbers-with-voice-connector-group \  
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901 \  
  --e164-phone-numbers "+12065550100" "+12065550101" \  
  --force-associate
```

Output:

```
{  
  "PhoneNumberErrors": []
```

```
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di Amazon Chime Voice Connector](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta AWS CLI Command [AssociatePhoneNumbersWithVoiceConnectorGroup](#)Reference.

## **associate-phone-numbers-with-voice-connector**

Il seguente esempio di codice mostra come utilizzare `associate-phone-numbers-with-voice-connector`.

### AWS CLI

Per associare numeri di telefono a un Amazon Chime Voice Connector

L'esempio seguente associa i numeri di telefono specificati a un Amazon Chime Voice Connector.

```
aws chime associate-phone-numbers-with-voice-connector \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --e164-phone-numbers " +12065550100" "+12065550101" \  
  --force-associate
```

Output:

```
{  
  "PhoneNumberErrors": []  
}
```

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [AssociatePhoneNumbersWithVoiceConnector](#)Reference.

## **associate-signin-delegate-groups-with-account**

Il seguente esempio di codice mostra come utilizzare `associate-signin-delegate-groups-with-account`.

## AWS CLI

Per associare gruppi di delegati all'accesso

L'associate-signin-delegate-groups-with-account esempio seguente associa il gruppo di delegati di accesso specificato all'account Amazon Chime specificato.

```
aws chime associate-signin-delegate-groups-with-account \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --signin-delegate-groups GroupName=my_users
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Gestione dell'accesso e delle autorizzazioni degli utenti](#) nella Guida all'amministrazione di Amazon Chime.

- Per API i dettagli, consulta AWS CLI Command [AssociateSigninDelegateGroupsWithAccount](#) Reference.

## batch-create-room-membership

Il seguente esempio di codice mostra come utilizzare batch-create-room-membership.

### AWS CLI

Per creare più iscrizioni a camere

L'batch-create-room-membership esempio seguente aggiunge più utenti a una chat room come membri della chat room. Assegna inoltre ruoli di amministratore e membro agli utenti.

```
aws chime batch-create-room-membership \  
  --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"
```

Output:

```
{
```

```

"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": []
}

```

Per ulteriori informazioni, consulta [Creare una chat room](#) nella Guida per l'utente di Amazon Chime.

- Per API i dettagli, consulta [BatchCreateRoomMembership AWS CLI Command Reference](#).

## batch-delete-phone-number

Il seguente esempio di codice mostra come utilizzare `batch-delete-phone-number`.

### AWS CLI

Per eliminare più numeri di telefono

L'esempio seguente elimina tutti i numeri di telefono specificati.

```

aws chime batch-delete-phone-number \
  --phone-number-ids "%2B12065550100" "%2B12065550101"

```

Questo comando non produce alcun output. Output:

```

{
  "PhoneNumberErrors": []
}

```

Per ulteriori informazioni, consulta [Lavorare con i numeri di telefono](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [BatchDeletePhoneNumber AWS CLI Command Reference](#).

## batch-suspend-user

Il seguente esempio di codice mostra come utilizzare `batch-suspend-user`.

### AWS CLI

Per sospendere più utenti

L'`batch-suspend-user` esempio seguente sospende gli utenti elencati dall'account Amazon Chime specificato.

```
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"
```

Output:

```
{  
  "UserErrors": []  
}
```

- Per API i dettagli, consulta AWS CLI Command [BatchSuspendUser](#) Reference.

## batch-unsuspend-user

Il seguente esempio di codice mostra come utilizzare `batch-unsuspend-user`.

### AWS CLI

Per annullare la sospensione di più utenti

L'`batch-unsuspend-user` esempio seguente rimuove qualsiasi sospensione precedente per gli utenti elencati sull'account Amazon Chime specificato.

```
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"
```

Output:

```
{
  "UserErrors": []
}
```

- Per API i dettagli, consulta [BatchUnsuspendUser AWS CLI Command Reference](#).

## batch-update-phone-number

Il seguente esempio di codice mostra come utilizzare `batch-update-phone-number`.

AWS CLI

Per aggiornare più tipi di prodotto con numeri di telefono contemporaneamente

L'`batch-update-phone-number` esempio seguente aggiorna i tipi di prodotto per tutti i numeri di telefono specificati.

```
aws chime batch-update-phone-number \
  --update-phone-number-request-items PhoneNumberId=
%2B12065550100,ProductType=BusinessCalling PhoneNumberId=
%2B12065550101,ProductType=BusinessCalling
```

Output:

```
{
  "PhoneNumberErrors": []
}
```

Per aggiornare più numeri di telefono, chiamare i nomi contemporaneamente

L'`batch-update-phone-number` esempio seguente aggiorna i nomi chiamanti per tutti i numeri di telefono specificati.

```
aws chime batch-update-phone-number \
  --update-phone-number-request-items PhoneNumberId=
%2B14013143874,CallingName=phonenumber1 PhoneNumberId=
%2B14013144061,CallingName=phonenumber2
```

**Output:**

```
{
  "PhoneNumberErrors": []
}
```

Per ulteriori informazioni, consulta [Lavorare con i numeri di telefono](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [BatchUpdatePhoneNumber AWS CLI Command Reference](#).

**batch-update-user**

Il seguente esempio di codice mostra come utilizzare `batch-update-user`.

**AWS CLI**

Per aggiornare più utenti con un solo comando

L'esempio seguente aggiorna il `LicenseType` per ciascuno degli utenti elencati nell'account Amazon Chime specificato.

```
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"
```

**Output:**

```
{
  "UserErrors": []
}
```

- Per API i dettagli, consulta [BatchUpdateUser AWS CLI Command Reference](#).

**create-account**

Il seguente esempio di codice mostra come utilizzare `create-account`.

## AWS CLI

Per creare un account

L'create-accountesempio seguente crea un account Amazon Chime con l'account dell' AWS amministratore.

```
aws chime create-account \  
  --name MyChimeAccount
```

Output:

```
{  
  "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"  
      },  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Getting Started](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [CreateAccount AWS CLI](#) Command Reference.

## create-bot

Il seguente esempio di codice mostra come utilizzare create-bot.

## AWS CLI

Per creare un bot Amazon Chime



L'`create-botesempio` seguente crea un bot per l'account Amazon Chime Enterprise specificato.

```
aws chime create-bot \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --display-name "myBot" \  
  --domain "example.com"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Integrazione di un chat bot con Amazon Chime nella Amazon Chime Developer Guide](#).

- Per API i dettagli, consulta AWS CLI Command [CreateBot](#) Reference.

## create-phone-number-order

Il seguente esempio di codice mostra come utilizzare `create-phone-number-order`.

### AWS CLI

Per creare un ordine di numeri di telefono

L'`create-phone-number-order` seguente crea un ordine di numeri di telefono per i numeri di telefono specificati.

```
aws chime create-phone-number-order \  
  --product-type VoiceConnector \  
  --e164-phone-numbers " +12065550100" "+12065550101" "+12065550102"
```

**Output:**

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Lavorare con i numeri di telefono](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [CreatePhoneNumberOrder AWS CLI Command Reference](#).

**create-proxy-session**

Il seguente esempio di codice mostra come utilizzare `create-proxy-session`.

**AWS CLI**

Per creare una sessione proxy

L'esempio seguente crea una sessione proxy con voce e SMS funzionalità.

```
aws chime create-proxy-session \
```

```
--voice-connector-id abcdef1ghij2klmno3pqr4 \  
--participant-phone-numbers "+14015550101" "+12065550100" \  
--capabilities "Voice" "SMS"
```

Output:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Proxy Phone Sessions](#) nella Amazon Chime Developer Guide.

- Per API i dettagli, consulta [CreateProxySession AWS CLI Command Reference](#).

## create-room-membership

Il seguente esempio di codice mostra come utilizzare `create-room-membership`.

### AWS CLI

Per creare un abbonamento a una sala

L'`create-room-membership` seguente aggiunge l'utente specificato alla chat room come membro della chat room.

```
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
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Creare una chat room](#) nella Guida per l'utente di Amazon Chime.

- Per API i dettagli, consulta [CreateRoomMembership AWS CLI Command Reference](#).

## **create-room**

Il seguente esempio di codice mostra come utilizzare `create-room`.

### AWS CLI

Per creare una chat room

L'`create-room` seguente crea una chat room per l'account Amazon Chime specificato.

```
aws chime create-room \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --room-name my-room
```

```
--account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
--name chatRoom
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Creare una chat room](#) nella Guida per l'utente di Amazon Chime.

- Per API i dettagli, consulta [CreateRoom AWS CLI Command Reference](#).

## create-user

Il seguente esempio di codice mostra come utilizzare `create-user`.

### AWS CLI

Per creare un profilo utente per un dispositivo condiviso

L'esempio seguente crea un profilo di dispositivo condiviso per l'indirizzo e-mail specificato.

```
aws chime create-user \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --email roomdevice@example.com \  
  --user-type SharedDevice
```

Output:

```
{  
  "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
    }
  }
}

```

Per ulteriori informazioni, consulta [Preparazione per l'installazione](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [CreateUser AWS CLI Command Reference](#).

## create-voice-connector-group

Il seguente esempio di codice mostra come utilizzare `create-voice-connector-group`.

### AWS CLI

Per creare un gruppo Amazon Chime Voice Connector

L'esempio seguente crea un gruppo Amazon Chime Voice Connector che include lo specifico Amazon Chime Voice Connector.

```

aws chime create-voice-connector-group \
  --name myGroup \
  --voice-connector-items VoiceConnectorId=abcdefghijklmno3pqr4,Priority=2

```

Output:

```

{
  "VoiceConnectorGroup": {
    "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jkl18901",
    "Name": "myGroup",
    "VoiceConnectorItems": [],
    "CreatedTimestamp": "2019-09-18T16:38:34.734Z",
  }
}

```

```

    "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"
  }
}

```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di Amazon Chime Voice Connector](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateVoiceConnectorGroupReference](#).

## create-voice-connector

Il seguente esempio di codice mostra come utilizzare `create-voice-connector`.

### AWS CLI

Per creare un Amazon Chime Voice Connector

L'esempio seguente crea un Amazon Chime Voice Connector nella AWS regione specificata, con la crittografia abilitata.

```

aws chime create-voice-connector \
  --name newVoiceConnector \
  --aws-region us-west-2 \
  --require-encryption

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [CreateVoiceConnector](#)Reference.

## **delete-account**

Il seguente esempio di codice mostra come utilizzare `delete-account`.

### AWS CLI

Per eliminare un account

L'`delete-account`esempio seguente elimina l'account specificato.

```
aws chime delete-account --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione dell'account](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteAccount](#)Reference.

## **delete-phone-number**

Il seguente esempio di codice mostra come utilizzare `delete-phone-number`.

### AWS CLI

Per eliminare un numero di telefono

L'`delete-phone-number`esempio seguente sposta il numero di telefono specificato nella coda di eliminazione.

```
aws chime delete-phone-number \  
  --phone-number-id " +12065550100"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lavorare con i numeri di telefono](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [DeletePhoneNumber AWS CLI](#)Command Reference.



## delete-proxy-session

Il seguente esempio di codice mostra come utilizzare `delete-proxy-session`.

### AWS CLI

Per eliminare una sessione proxy

L'`delete-proxy-session` esempio seguente elimina la sessione proxy specificata.

```
aws chime delete-proxy-session \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk567891
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Proxy Phone Sessions](#) nella Amazon Chime Developer Guide.

- Per API i dettagli, consulta [DeleteProxySession AWS CLI Command Reference](#).

## delete-room-membership

Il seguente esempio di codice mostra come utilizzare `delete-room-membership`.

### AWS CLI

Per rimuovere un utente come membro di una chat room

L'`delete-room-membership` esempio seguente rimuove il membro specificato dalla chat room specificata.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creare una chat room](#) nella Guida per l'utente di Amazon Chime.

- Per API i dettagli, consulta [DeleteRoomMembership AWS CLI Command Reference](#).

## delete-room

Il seguente esempio di codice mostra come utilizzare `delete-room`.

### AWS CLI

Per eliminare una chat room

L'esempio seguente elimina la chat room specificata e rimuove le appartenenze alla chat room.

```
aws chime delete-room \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creare una chat room](#) nella Guida per l'utente di Amazon Chime.

- Per API i dettagli, consulta [DeleteRoom AWS CLI Command Reference](#).

## delete-voice-connector-group

Il seguente esempio di codice mostra come utilizzare `delete-voice-connector-group`.

### AWS CLI

titolo

L'esempio seguente elimina il gruppo Amazon Chime Voice Connector specificato.

```
aws chime delete-voice-connector-group \  
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lavorare con i gruppi di Amazon Chime Voice Connector](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteVoiceConnectorGroupReference](#).

## **delete-voice-connector-origination**

Il seguente esempio di codice mostra come utilizzare `delete-voice-connector-origination`.

### AWS CLI

Per eliminare le impostazioni di origine

L'`delete-voice-connector-origination` esempio seguente elimina l'host, la porta, il protocollo, la priorità e il peso di origine dal Amazon Chime Voice Connector specificato.

```
aws chime delete-voice-connector-origination \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [DeleteVoiceConnectorOriginationReference](#).

## **delete-voice-connector-proxy**

Il seguente esempio di codice mostra come utilizzare `delete-voice-connector-proxy`.

### AWS CLI

Per eliminare una configurazione proxy

L'`delete-voice-connector-proxy` esempio seguente elimina la configurazione del proxy dal tuo Amazon Chime Voice Connector.

```
aws chime delete-voice-connector-proxy \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Proxy Phone Sessions](#) nella Amazon Chime Developer Guide.

- Per API i dettagli, consulta [DeleteVoiceConnectorProxy AWS CLI Command Reference](#).

## **delete-voice-connector-streaming-configuration**

Il seguente esempio di codice mostra come utilizzare `delete-voice-connector-streaming-configuration`.

### AWS CLI

Per eliminare una configurazione di streaming

L'`delete-voice-connector-streaming-configuration` esempio seguente elimina la configurazione di streaming per il connettore Amazon Chime Voice specificato.

```
aws chime delete-voice-connector-streaming-configuration \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Streaming dei dati di Amazon Chime Voice Connector su Kinesis](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta Command [DeleteVoiceConnectorStreamingConfiguration Reference](#) AWS CLI .

## **delete-voice-connector-termination-credentials**

Il seguente esempio di codice mostra come utilizzare `delete-voice-connector-termination-credentials`.

### AWS CLI

Per eliminare le credenziali di terminazione

L'`delete-voice-connector-termination-credentials` esempio seguente elimina le credenziali di terminazione per il nome utente specificato e Amazon Chime Voice Connector.

```
aws chime delete-voice-connector-termination-credentials \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --usernames "jdoe"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [DeleteVoiceConnectorTerminationCredentials](#)Reference.

## **delete-voice-connector-termination**

Il seguente esempio di codice mostra come utilizzare `delete-voice-connector-termination`.

### AWS CLI

Per eliminare le impostazioni di terminazione

L'`delete-voice-connector-termination`esempio seguente elimina le impostazioni di terminazione per l'Amazon Chime Voice Connector specificato.

```
aws chime delete-voice-connector-termination \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [DeleteVoiceConnectorTermination](#)Reference.

## **delete-voice-connector**

Il seguente esempio di codice mostra come utilizzare `delete-voice-connector`.

### AWS CLI

Per eliminare un Amazon Chime Voice Connector

L'`delete-voice-connector`esempio seguente esegue questa operazione

```
aws chime delete-voice-connector \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [DeleteVoiceConnectorReference](#).

## **disassociate-phone-number-from-user**

Il seguente esempio di codice mostra come utilizzare `disassociate-phone-number-from-user`.

### AWS CLI

Per dissociare un numero di telefono da un utente

L'`disassociate-phone-number-from-user` esempio seguente dissocia un numero di telefono dall'utente specificato.

```
aws chime disassociate-phone-number-from-user \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Gestione dei numeri di telefono degli utenti](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [DisassociatePhoneNumberFromUser AWS CLI Command Reference](#).

## **disassociate-phone-numbers-from-voice-connector-group**

Il seguente esempio di codice mostra come utilizzare `disassociate-phone-numbers-from-voice-connector-group`.

### AWS CLI

Per dissociare i numeri di telefono da un gruppo Amazon Chime Voice Connector

L'`disassociate-phone-numbers-from-voice-connector-group` esempio seguente dissocia i numeri di telefono specificati da un gruppo Amazon Chime Voice Connector.

```
aws chime disassociate-phone-numbers-from-voice-connector-group \  
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jkl8901 \  
  --e164-phone-numbers "+12065550100" "+12065550101"
```

Output:

```
{  
  "PhoneNumberErrors": []  
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di Amazon Chime Voice Connector](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta AWS CLI Command [DisassociatePhoneNumbersFromVoiceConnectorGroup](#)Reference.

## disassociate-phone-numbers-from-voice-connector

Il seguente esempio di codice mostra come utilizzare `disassociate-phone-numbers-from-voice-connector`.

AWS CLI

Per dissociare i numeri di telefono da un Amazon Chime Voice Connector

L'`disassociate-phone-numbers-from-voice-connector` esempio seguente dissocia i numeri di telefono specificati da un Amazon Chime Voice Connector.

```
aws chime disassociate-phone-numbers-from-voice-connector \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --e164-phone-numbers "+12065550100" "+12065550101"
```

Output:

```
{  
  "PhoneNumberErrors": []  
}
```

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [DisassociatePhoneNumbersFromVoiceConnector](#)Reference.

## **disassociate-signin-delegate-groups-from-account**

Il seguente esempio di codice mostra come utilizzare `disassociate-signin-delegate-groups-from-account`.

### AWS CLI

Per dissociare i gruppi di delegati all'accesso

L'`disassociate-signin-delegate-groups-from-account`esempio seguente dissocia il gruppo di delegati di accesso specificato dall'`account` Amazon Chime specificato.

```
aws chime disassociate-signin-delegate-groups-from-account \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --group-names "my_users"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Gestione dell'accesso e delle autorizzazioni degli utenti](#) nella Guida all'amministrazione di Amazon Chime.

- Per API i dettagli, consulta AWS CLI Command [DisassociateSigninDelegateGroupsFromAccount](#)Reference.

## **get-account-settings**

Il seguente esempio di codice mostra come utilizzare `get-account-settings`.

### AWS CLI

Per recuperare le impostazioni di un `account`

L'`get-account-settings`esempio seguente recupera le impostazioni dell'`account` per l'`account` specificato.

```
aws chime get-account-settings --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Output:



```
{
  "AccountSettings": {
    "DisableRemoteControl": false,
    "EnableDialOut": false
  }
}
```

Per ulteriori informazioni, consulta [Managing Your Amazon Chime Accounts nella Amazon Chime Administration Guide](#).

- Per API i dettagli, consulta AWS CLI Command [GetAccountSettingsReference](#).

## get-account

Il seguente esempio di codice mostra come utilizzare `get-account`.

### AWS CLI

Per recuperare i dettagli di un account

L'`get-account` seguente recupera i dettagli per l'account Amazon Chime specificato.

```
aws chime get-account \
  --account-id a1b2c3d4-5678-90ab-cdef-1111EXAMPLE
```

Output:

```
{
  "Account": {
    "AwsAccountId": "111122223333",
    "AccountId": "a1b2c3d4-5678-90ab-cdef-1111EXAMPLE",
    "Name": "EnterpriseDirectory",
    "AccountType": "EnterpriseDirectory",
    "CreatedTimestamp": "2018-12-20T18:38:02.181Z",
    "DefaultLicense": "Pro",
    "SupportedLicenses": [
      "Basic",
      "Pro"
    ],
    "SigninDelegateGroups": [
      {
        "GroupName": "myGroup"
      }
    ]
  }
}
```

```

    },
  ]
}
}

```

Per ulteriori informazioni, consulta [Managing Your Amazon Chime Accounts nella Amazon Chime Administration Guide](#).

- Per API i dettagli, consulta AWS CLI Command [GetAccountReference](#).

## get-bot

Il seguente esempio di codice mostra come utilizzare `get-bot`.

### AWS CLI

Per recuperare dettagli su un bot

L'`get-bot`esempio seguente visualizza i dettagli per il bot specificato.

```

aws chime get-bot \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Update Chat Bots](#) nella Amazon Chime Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetBotReference](#).

## get-global-settings

Il seguente esempio di codice mostra come utilizzare `get-global-settings`.

### AWS CLI

Per ottenere impostazioni globali

L'esempio seguente recupera i nomi dei bucket S3 utilizzati per archiviare i record dei dettagli delle chiamate per Amazon Chime Business Calling e Amazon Chime Voice Connectors associati all'account dell'amministratore. AWS

```
aws chime get-global-settings
```

Output:

```
{
  "BusinessCalling": {
    "CdrBucket": "s3bucket"
  },
  "VoiceConnector": {
    "CdrBucket": "s3bucket"
  }
}
```

Per ulteriori informazioni, consulta [Managing Global Settings](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [GetGlobalSettings AWS CLI Command Reference](#).

## get-phone-number-order

Il seguente esempio di codice mostra come utilizzare `get-phone-number-order`.

### AWS CLI

Per ottenere i dettagli relativi all'ordine di un numero di telefono

L'esempio seguente visualizza i dettagli dell'ordine del numero di telefono specificato.

```
aws chime get-phone-number-order \
```

```
--phone-number-order-id abc12345-de67-89f0-123g-h45i678j9012
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Lavorare con i numeri di telefono](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [GetPhoneNumberOrder AWS CLI Command Reference](#).

## get-phone-number-settings

Il seguente esempio di codice mostra come utilizzare `get-phone-number-settings`.

AWS CLI

Per recuperare un nome di chiamata in uscita

L'esempio seguente recupera il nome di chiamata in uscita predefinito per l'account dell'utente chiamante. AWS

**aws chime get-phone-number-settings**

Questo comando non produce alcun output. Output:

```
{
  "CallingName": "myName",
  "CallingNameUpdatedTimestamp": "2019-10-28T18:56:42.911Z"
}
```

Per ulteriori informazioni, consulta [Lavorare con i numeri di telefono](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [GetPhoneNumberSettings AWS CLI Command Reference](#).

**get-phone-number**

Il seguente esempio di codice mostra come utilizzare `get-phone-number`.

## AWS CLI

Per ottenere i dettagli del numero di telefono

L'`get-phone-number` esempio seguente visualizza i dettagli del numero di telefono specificato.

```
aws chime get-phone-number \
  --phone-number-id +12065550100
```

Output:

```
{
  "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"
}
}

```

Per ulteriori informazioni, consulta [Lavorare con i numeri di telefono](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [GetPhoneNumber AWS CLI](#) Command Reference.

## get-proxy-session

Il seguente esempio di codice mostra come utilizzare `get-proxy-session`.

### AWS CLI

Per ottenere i dettagli della sessione proxy

L'`get-proxy-session` seguente elenca i dettagli della sessione proxy specificata.

```

aws chime get-proxy-session \
  --voice-connector-id abcdef1ghij2klmno3pqr4 \
  --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk567891

```

Output:

```

{
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Proxy Phone Sessions](#) nella Amazon Chime Developer Guide.

- Per API i dettagli, consulta [GetProxySession AWS CLI Command Reference](#).

## get-room

Il seguente esempio di codice mostra come utilizzare `get-room`.

### AWS CLI

Per ottenere i dettagli su una chat room

L'`get-room` esempio seguente mostra i dettagli sulla chat room specificata.

```

aws chime get-room \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j

```

Output:

```

{
  "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"
}
}
```

Per ulteriori informazioni, consulta [Creare una chat room](#) nella Guida per l'utente di Amazon Chime.

- Per API i dettagli, consulta [GetRoom AWS CLI Command Reference](#).

## get-user-settings

Il seguente esempio di codice mostra come utilizzare `get-user-settings`.

### AWS CLI

Per recuperare le impostazioni dell'utente

L'`get-user-settings` seguente visualizza le impostazioni utente specificate.

```
aws chime get-user-settings \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Output:

```
{
  "UserSettings": {
    "Telephony": {
      "InboundCalling": true,
      "OutboundCalling": true,
      "SMS": true
    }
  }
}
```

Per ulteriori informazioni, consulta la sezione [Gestione dei numeri di telefono degli utenti](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [GetUserSettings AWS CLI Command Reference](#).



## get-user

Il seguente esempio di codice mostra come utilizzare `get-user`.

### AWS CLI

Per ottenere dettagli su un utente

L'`get-user` seguente recupera i dettagli per l'utente specificato.

```
aws chime get-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Managing Users](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [GetUser AWS CLI Command Reference](#).

## get-voice-connector-group

Il seguente esempio di codice mostra come utilizzare `get-voice-connector-group`.

## AWS CLI

Per ottenere dettagli su un gruppo Amazon Chime Voice Connector

L'get-voice-connector-groupesempio seguente mostra i dettagli per il gruppo Amazon Chime Voice Connector specificato.

```
aws chime get-voice-connector-group \  
--voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901
```

Output:

```
{  
  "VoiceConnectorGroup": {  
    "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jk18901",  
    "Name": "myGroup",  
    "VoiceConnectorItems": [],  
    "CreatedTimestamp": "2019-09-18T16:38:34.734Z",  
    "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"  
  }  
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di Amazon Chime Voice Connector](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta AWS CLI Command [GetVoiceConnectorGroupReference](#).

## get-voice-connector-logging-configuration

Il seguente esempio di codice mostra come utilizzare get-voice-connector-logging-configuration.

## AWS CLI

Per ottenere i dettagli di configurazione della registrazione

L'get-voice-connector-logging-configurationesempio seguente recupera i dettagli di configurazione della registrazione per l'Amazon Chime Voice Connector specificato.

```
aws chime get-voice-connector-logging-configuration \  
--voice-connector-id abcdef1ghij2klmno3pqr4
```

**Output:**

```
{
  "LoggingConfiguration": {
    "EnableSIPLogs": true
  }
}
```

Per ulteriori informazioni, consulta [Streaming dei contenuti multimediali di Amazon Chime Voice Connector su Kinesis](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta Command [GetVoiceConnectorLoggingConfiguration](#) Reference AWS CLI .

**get-voice-connector-origination**

Il seguente esempio di codice mostra come utilizzare `get-voice-connector-origination`.

**AWS CLI**

Per recuperare le impostazioni di origine

L'`get-voice-connector-origination` esempio seguente recupera l'host, la porta, il protocollo, la priorità e il peso di origine per l'Amazon Chime Voice Connector specificato.

```
aws chime get-voice-connector-origination \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

**Output:**

```
{
  "Origination": {
    "Routes": [
      {
        "Host": "10.24.34.0",
        "Port": 1234,
        "Protocol": "TCP",
        "Priority": 1,
        "Weight": 5
      }
    ],
    "Disabled": false
  }
}
```

```
}  
}
```

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [GetVoiceConnectorOrigination](#) Reference.

## get-voice-connector-proxy

Il seguente esempio di codice mostra come utilizzare `get-voice-connector-proxy`.

### AWS CLI

Per ottenere i dettagli sulla configurazione del proxy

L'esempio seguente ottiene i dettagli di configurazione del proxy per Amazon Chime Voice Connector.

```
aws chime get-voice-connector-proxy \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Output:

```
{  
  "Proxy": {  
    "DefaultSessionExpiryMinutes": 60,  
    "Disabled": false,  
    "PhoneNumberCountries": [  
      "US"  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Proxy Phone Sessions](#) nella Amazon Chime Developer Guide.

- Per API i dettagli, consulta [GetVoiceConnectorProxy AWS CLI Command Reference](#).

## get-voice-connector-streaming-configuration

Il seguente esempio di codice mostra come utilizzare `get-voice-connector-streaming-configuration`.

## AWS CLI

Per ottenere i dettagli della configurazione dello streaming

L'`get-voice-connector-streaming-configuration`esempio seguente ottiene i dettagli di configurazione dello streaming per il connettore Amazon Chime Voice specificato.

```
aws chime get-voice-connector-streaming-configuration \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Output:

```
{  
  "StreamingConfiguration": {  
    "DataRetentionInHours": 24,  
    "Disabled": false  
  }  
}
```

Per ulteriori informazioni, consulta [Streaming dei dati di Amazon Chime Voice Connector su Kinesis](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta Command [GetVoiceConnectorStreamingConfiguration](#)Reference AWS CLI .

## `get-voice-connector-termination-health`

Il seguente esempio di codice mostra come utilizzare `get-voice-connector-termination-health`.

## AWS CLI

Per recuperare i dettagli relativi allo stato di salute in caso di interruzione

L'`get-voice-connector-termination-health`esempio seguente recupera i dettagli sullo stato di terminazione per l'Amazon Chime Voice Connector specificato.

```
aws chime get-voice-connector-termination-health \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

**Output:**

```
{
  "TerminationHealth": {
    "Timestamp": "Fri Aug 23 16:45:55 UTC 2019",
    "Source": "10.24.34.0"
  }
}
```

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [GetVoiceConnectorTerminationHealth](#) Reference.

**get-voice-connector-termination**

Il seguente esempio di codice mostra come utilizzare `get-voice-connector-termination`.

**AWS CLI**

Per recuperare le impostazioni di terminazione

L'`get-voice-connector-termination` esempio seguente recupera le impostazioni di terminazione per l'Amazon Chime Voice Connector specificato.

```
aws chime get-voice-connector-termination \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Questo comando non produce alcun output. Output:

```
{
  "Termination": {
    "CpsLimit": 1,
    "DefaultPhoneNumber": "+12065550100",
    "CallingRegions": [
      "US"
    ],
    "CidrAllowedList": [
      "10.24.34.0/23"
    ],
  },
}
```

```
    "Disabled": false
  }
}
```

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [GetVoiceConnectorTermination](#)Reference.

## get-voice-connector

Il seguente esempio di codice mostra come utilizzare `get-voice-connector`.

### AWS CLI

Per ottenere dettagli su un Amazon Chime Voice Connector

L'`get-voice-connector` esempio seguente mostra i dettagli del connettore Amazon Chime Voice specificato.

```
aws chime get-voice-connector \
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [GetVoiceConnector](#)Reference.

## invite-users

Il seguente esempio di codice mostra come utilizzare `invite-users`.

### AWS CLI

Per invitare gli utenti a iscriversi ad Amazon Chime

L'esempio seguente invia un'e-mail per invitare un utente all'account Amazon Chime specificato.

```
aws chime invite-users \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-email-list "alejandror@example.com" "janed@example.com"
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Invitare e sospendere gli utenti](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta Command [InviteUsers](#) Reference AWS CLI .

## list-accounts

Il seguente esempio di codice mostra come utilizzare `list-accounts`.



## AWS CLI

Per ottenere un elenco di account

L'`list-account`sempio seguente recupera un elenco degli account Amazon Chime nell'account dell'amministratore AWS .

```
aws chime list-accounts
```

Output:

```
{
  "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"
  },
]
}

```

Per ulteriori informazioni, consulta [Managing Your Amazon Chime Accounts nella Amazon Chime Administration Guide](#).

- Per API i dettagli, consulta AWS CLI Command [ListAccounts](#) Reference.

## list-bots

Il seguente esempio di codice mostra come utilizzare `list-bots`.

### AWS CLI

Per recuperare un elenco di bot

L'`list-bots` esempio seguente elenca i bot associati all'account Amazon Chime Enterprise specificato.

```

aws chime list-bots \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Usare i chat bot con Amazon Chime nella Amazon Chime Developer Guide](#).

- Per API i dettagli, consulta Command [ListBots](#)Reference AWS CLI .

## **list-phone-number-orders**

Il seguente esempio di codice mostra come utilizzare `list-phone-number-orders`.

### AWS CLI

Per elencare gli ordini relativi ai numeri di telefono

L'`list-phone-number-orders`esempio seguente elenca gli ordini di numeri di telefono associati all'account dell'amministratore di Amazon Chime.

```
aws chime list-phone-number-orders
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Lavorare con i numeri di telefono](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [ListPhoneNumberOrders AWS CLI Command Reference](#).

## list-phone-numbers

Il seguente esempio di codice mostra come utilizzare `list-phone-numbers`.

### AWS CLI

Per elencare i numeri di telefono di un account Amazon Chime

L'`list-phone-numbers` esempio seguente elenca i numeri di telefono associati all'account Amazon Chime dell'amministratore.

```
aws chime list-phone-numbers
```

Questo comando non produce alcun output. Output:

```
{
  "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"
}
]
}

```

Per ulteriori informazioni, consulta [Lavorare con i numeri di telefono](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [ListPhoneNumbers AWS CLI](#) Command Reference.

## list-proxy-sessions

Il seguente esempio di codice mostra come utilizzare `list-proxy-sessions`.

### AWS CLI

Per elencare le sessioni proxy

L'`list-proxy-session`esempio seguente elenca le sessioni proxy per Amazon Chime Voice Connector.

```

aws chime list-proxy-sessions \
  --voice-connector-id abcdef1ghij2klmno3pqr4

```

Output:

```

{
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Proxy Phone Sessions](#) nella Amazon Chime Developer Guide.

- Per API i dettagli, consulta [ListProxySessions AWS CLI Command Reference](#).

## list-room-memberships

Il seguente esempio di codice mostra come utilizzare `list-room-memberships`.

### AWS CLI

Per elencare gli abbonamenti alle sale

L'esempio seguente visualizza un elenco dei dettagli di iscrizione per la chat room specificata.

```

aws chime list-room-memberships \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j

```

Output:

```

{
  "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"
}
]
}

```

Per ulteriori informazioni, consulta [Creare una chat room](#) nella Guida per l'utente di Amazon Chime.

- Per API i dettagli, consulta [ListRoomMemberships AWS CLI Command Reference](#).

## list-rooms

Il seguente esempio di codice mostra come utilizzare `list-rooms`.

### AWS CLI

Per elencare le chat room

L'esempio seguente visualizza un elenco di chat room nell'account specificato. L'elenco viene filtrato solo in base alle chat room a cui appartiene il membro specificato.

```

aws chime list-rooms \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \

```



```
--member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Creare una chat room](#) nella Guida per l'utente di Amazon Chime.

- Per API i dettagli, consulta [ListRooms AWS CLI Command Reference](#).

## list-users

Il seguente esempio di codice mostra come utilizzare `list-users`.

### AWS CLI

Per elencare gli utenti di un account

L'`list-users` esempio seguente elenca gli utenti per l'account Amazon Chime specificato.

```
aws chime list-users --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Output:

```
{
  "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
    }
  }
}
```

```
    }  
  }  
]  
}
```

Per ulteriori informazioni, consulta [Managing Users](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [ListUsers AWS CLI Command Reference](#).

## **list-voice-connector-groups**

Il seguente esempio di codice mostra come utilizzare `list-voice-connector-groups`.

### AWS CLI

Per elencare i gruppi Amazon Chime Voice Connector per un account Amazon Chime

L'`list-voice-connector-groups` seguente elenca i gruppi Amazon Chime Voice Connector associati all'account Amazon Chime dell'amministratore.

```
aws chime list-voice-connector-groups
```

Output:

```
{  
  "VoiceConnectorGroups": [  
    {  
      "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jk18901",  
      "Name": "myGroup",  
      "VoiceConnectorItems": [],  
      "CreatedTimestamp": "2019-09-18T16:38:34.734Z",  
      "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di Amazon Chime Voice Connector](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta AWS CLI Command [ListVoiceConnectorGroups Reference](#).

## list-voice-connector-termination-credentials

Il seguente esempio di codice mostra come utilizzare `list-voice-connector-termination-credentials`.

### AWS CLI

Per recuperare un elenco di credenziali di terminazione

L'`list-voice-connector-termination-credentials` seguente recupera un elenco delle credenziali di terminazione per l'Amazon Chime Voice Connector specificato.

```
aws chime list-voice-connector-termination-credentials \  
  --voice-connector-id abcdef1ghij2klmno3pqr4
```

Questo comando non produce alcun output. Output:

```
{  
  "Usernames": [  
    "jdoe"  
  ]  
}
```

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [ListVoiceConnectorTerminationCredentialsReference](#).

## list-voice-connectors

Il seguente esempio di codice mostra come utilizzare `list-voice-connectors`.

### AWS CLI

Per elencare i connettori vocali Amazon Chime per un account

L'`list-voice-connectors` seguente elenca i connettori vocali Amazon Chime associati all'account del chiamante.

```
aws chime list-voice-connectors
```

**Output:**

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [ListVoiceConnectors](#) Reference.

**logout-user**

Il seguente esempio di codice mostra come utilizzare `logout-user`.

**AWS CLI**

Per disconnettere un utente

L'`logout-user` esempio seguente disconnette l'utente specificato.

```
aws chime logout-user \
  --account-id a1b2c3d4-5678-90ab-cdef-1111EXAMPLE \
  --user-id a1b2c3d4-5678-90ab-cdef-2222EXAMPLE
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [LogoutUser](#) in AWS CLI Command Reference.

## **put-voice-connector-logging-configuration**

Il seguente esempio di codice mostra come utilizzare `put-voice-connector-logging-configuration`.

### AWS CLI

Per aggiungere una configurazione di registrazione per un Amazon Chime Voice Connector

L'esempio seguente attiva la configurazione di SIP registrazione per il connettore Amazon Chime Voice specificato.

```
aws chime put-voice-connector-logging-configuration \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --logging-configuration EnableSIPLogs=true
```

Output:

```
{  
  "LoggingConfiguration": {  
    "EnableSIPLogs": true  
  }  
}
```

Per ulteriori informazioni, consulta [Streaming dei contenuti multimediali di Amazon Chime Voice Connector su Kinesis](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta Command [PutVoiceConnectorLoggingConfiguration](#) Reference AWS CLI .

## **put-voice-connector-origination**

Il seguente esempio di codice mostra come utilizzare `put-voice-connector-origination`.

### AWS CLI

Per configurare le impostazioni di origine

L'`put-voice-connector-origination` seguente imposta l'host, la porta, il protocollo, la priorità e il peso di origine per lo specifico Amazon Chime Voice Connector.

```
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
```

Output:

```
{  
  "Origination": {  
    "Routes": [  
      {  
        "Host": "10.24.34.0",  
        "Port": 1234,  
        "Protocol": "TCP",  
        "Priority": 1,  
        "Weight": 5  
      }  
    ],  
    "Disabled": false  
  }  
}
```

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [PutVoiceConnectorOrigination](#) Reference.

## **put-voice-connector-proxy**

Il seguente esempio di codice mostra come utilizzare `put-voice-connector-proxy`.

### AWS CLI

Per inserire una configurazione proxy

L'`put-voice-connector-proxy` seguente imposta una configurazione proxy per Amazon Chime Voice Connector.

```
aws chime put-voice-connector-proxy \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --proxy-configuration {}
```

```
--voice-connector-id abcdef1ghij2klmno3pqr4 \  
--default-session-expiry-minutes 60 \  
--phone-number-pool-countries "US"
```

Output:

```
{  
  "Proxy": {  
    "DefaultSessionExpiryMinutes": 60,  
    "Disabled": false,  
    "PhoneNumberCountries": [  
      "US"  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Proxy Phone Sessions](#) nella Amazon Chime Developer Guide.

- Per API i dettagli, consulta [PutVoiceConnectorProxy AWS CLI Command Reference](#).

## put-voice-connector-streaming-configuration

Il seguente esempio di codice mostra come utilizzare `put-voice-connector-streaming-configuration`.

AWS CLI

Per creare una configurazione di streaming

L'esempio seguente crea una configurazione di streaming per il connettore Amazon Chime Voice specificato. Abilita lo streaming multimediale da Amazon Chime Voice Connector ad Amazon Kinesis e imposta il periodo di conservazione dei dati su 24 ore.

```
aws chime put-voice-connector-streaming-configuration \  
--voice-connector-id abcdef1ghij2klmno3pqr4 \  
--streaming-configuration DataRetentionInHours=24,Disabled=false
```

Output:

```
{  
  "StreamingConfiguration": {
```



```
    "DataRetentionInHours": 24,  
    "Disabled": false  
  }  
}
```

Per ulteriori informazioni, consulta [Streaming dei dati di Amazon Chime Voice Connector su Kinesis](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta Command [PutVoiceConnectorStreamingConfigurationReference](#) AWS CLI .

## **put-voice-connector-termination-credentials**

Il seguente esempio di codice mostra come utilizzare `put-voice-connector-termination-credentials`.

AWS CLI

Per impostare le credenziali di terminazione

L'`put-voice-connector-termination-credentials` esempio seguente imposta le credenziali di terminazione per lo specifico Amazon Chime Voice Connector.

```
aws chime put-voice-connector-termination-credentials \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --credentials Username="jdoe",Password="XXXXXXXX"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [PutVoiceConnectorTerminationCredentialsReference](#).

## **put-voice-connector-termination**

Il seguente esempio di codice mostra come utilizzare `put-voice-connector-termination`.

AWS CLI

Per configurare le impostazioni di terminazione

L'`put-voice-connector-termination`esempio seguente imposta le regioni di chiamata e le impostazioni di terminazione dell'host IP consentite per l'Amazon Chime Voice Connector specificato.

```
aws chime put-voice-connector-termination \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --termination CallingRegions="US",CidrAllowedList="10.24.34.0/23",Disabled=false
```

Output:

```
{  
  "Termination": {  
    "CpsLimit": 0,  
    "CallingRegions": [  
      "US"  
    ],  
    "CidrAllowedList": [  
      "10.24.34.0/23"  
    ],  
    "Disabled": false  
  }  
}
```

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida all'amministrazione di Amazon Chime](#).

- Per API i dettagli, consulta AWS CLI Command [PutVoiceConnectorTermination](#)Reference.

## **regenerate-security-token**

Il seguente esempio di codice mostra come utilizzare `regenerate-security-token`.

AWS CLI

Per rigenerare un token di sicurezza

L'`regenerate-security-token`esempio seguente rigenera il token di sicurezza per il bot specificato.

```
aws chime regenerate-security-token \  
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
  --bot-id 12345678-9012-3456-7890-123456789012
```

```
--bot-id 123abcd4-5ef6-789g-0h12-34j56789012k
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Authenticate Chat Bot Requests](#) nella Amazon Chime Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [RegenerateSecurityToken](#) Reference.

## reset-personal-pin

Il seguente esempio di codice mostra come utilizzare `reset-personal-pin`.

AWS CLI

Per reimpostare la riunione personale di un utente PIN

L'`reset-personal-pin` seguente reimposta la riunione PIN personale dell'utente specificato.

```
aws chime reset-personal-pin \
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Changing Personal Meeting PINs](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [ResetPersonalPin AWS CLI Command Reference](#).

## restore-phone-number

Il seguente esempio di codice mostra come utilizzare `restore-phone-number`.

### AWS CLI

Per ripristinare un numero di telefono

L'esempio seguente ripristina il numero di telefono specificato dalla coda di eliminazione.

```

aws chime restore-phone-number \
  --phone-number-id "+12065550100"

```

Output:

```

{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Lavorare con i numeri di telefono](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [RestorePhoneNumber AWS CLI](#) Command Reference.

## search-available-phone-numbers

Il seguente esempio di codice mostra come utilizzare `search-available-phone-numbers`.

### AWS CLI

Per cercare i numeri di telefono disponibili

L'esempio seguente cerca i numeri di telefono disponibili in base al prefisso.

```
aws chime search-available-phone-numbers \
  --area-code "206"
```

Output:

```
{
  "E164PhoneNumbers": [
    "+12065550100",
    "+12065550101",
    "+12065550102",
```

```
    "+12065550103",  
    "+12065550104",  
    "+12065550105",  
    "+12065550106",  
    "+12065550107",  
    "+12065550108",  
    "+12065550109",  
  ]  
}
```

Per ulteriori informazioni, consulta [Lavorare con i numeri di telefono](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [SearchAvailablePhoneNumbers AWS CLI](#) Command Reference.

## update-account-settings

Il seguente esempio di codice mostra come utilizzare `update-account-settings`.

### AWS CLI

Per aggiornare le impostazioni del tuo account

L'esempio seguente disabilita il controllo remoto degli schermi condivisi per l'account Amazon Chime specificato.

```
aws chime update-account-settings \  
  --account-id a1b2c3d4-5678-90ab-cdef-1111EXAMPLE \  
  --account-settings DisableRemoteControl=true
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta AWS CLI Command [UpdateAccountSettings](#) Reference.

## update-account

Il seguente esempio di codice mostra come utilizzare `update-account`.

### AWS CLI

Per aggiornare un account

L'esempio seguente aggiorna il nome di account specificato.

```
aws chime update-account \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --name MyAccountName
```

Output:

```
{  
  "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"  
      },  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Rinominare il tuo account](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateAccount](#)Reference.

## update-bot

Il seguente esempio di codice mostra come utilizzare `update-bot`.

### AWS CLI

Per aggiornare un bot

L'update-bot esempio seguente aggiorna lo stato del bot specificato per impedirne l'esecuzione.

```
aws chime update-bot \  
  --bot-id myBotId
```

```
--account-id 12a3456b-7c89-012d-3456-78901e23fg45 \  
--bot-id 123abcd4-5ef6-789g-0h12-34j56789012k \  
--disabled
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Update Chat Bots](#) nella Amazon Chime Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateBotReference](#).

## update-global-settings

Il seguente esempio di codice mostra come utilizzare `update-global-settings`.

AWS CLI

Per aggiornare le impostazioni globali

L'esempio seguente aggiorna il bucket S3 utilizzato per archiviare i record dei dettagli delle chiamate per Amazon Chime Business Calling e Amazon Chime Voice Connectors associati all'account dell'amministratore. AWS

```
aws chime update-global-settings \  
  --business-calling CdrBucket="s3bucket" \  
  --voice-connector CdrBucket="s3bucket"
```

Questo comando non produce alcun output.



Per ulteriori informazioni, consulta [Managing Global Settings](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [UpdateGlobalSettings AWS CLI Command Reference](#).

## update-phone-number-settings

Il seguente esempio di codice mostra come utilizzare `update-phone-number-settings`.

### AWS CLI

Per aggiornare il nome di una chiamata in uscita

L'esempio seguente aggiorna il nome predefinito per le chiamate in uscita per l'account dell'AWS amministratore.

```
aws chime update-phone-number-settings \  
  --calling-name "myName"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lavorare con i numeri di telefono](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [UpdatePhoneNumberSettings AWS CLI Command Reference](#).

## update-phone-number

Il seguente esempio di codice mostra come utilizzare `update-phone-number`.

### AWS CLI

Esempio 1: aggiornare il tipo di prodotto per un numero di telefono

L'esempio seguente aggiorna il tipo di prodotto del numero di telefono specificato.

```
aws chime update-phone-number \  
  --phone-number-id "+12065550100" \  
  --product-type "BusinessCalling"
```

Output:

```
{
  "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": "phonenumber1",
    "CreatedTimestamp": "2019-08-09T21:35:21.445Z",
    "UpdatedTimestamp": "2019-08-12T21:44:07.591Z"
  }
}
```

Esempio 2: aggiornare il nome per le chiamate in uscita per un numero di telefono

L'update-phone-number esempio seguente aggiorna il nome di chiamata in uscita per il numero di telefono specificato.

```
aws chime update-phone-number -- phone-number-id «+12065550100" --calling-name
«phonenumber2"
```

Output:

```
{
  "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"
}
}

```

Per ulteriori informazioni, consulta [Lavorare con i numeri di telefono](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [UpdatePhoneNumber AWS CLI Command Reference](#).

## update-proxy-session

Il seguente esempio di codice mostra come utilizzare `update-proxy-session`.

### AWS CLI

Per aggiornare una sessione proxy

L'esempio seguente aggiorna le funzionalità della sessione proxy.

```

aws chime update-proxy-session \
  --voice-connector-id abcdef1ghij2klmno3pqr4 \
  --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk567891 \
  --capabilities "Voice"

```

Output:

```

{
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Proxy Phone Sessions](#) nella Amazon Chime Developer Guide.

- Per API i dettagli, consulta [UpdateProxySession AWS CLI Command Reference](#).

## update-room-membership

Il seguente esempio di codice mostra come utilizzare `update-room-membership`.

### AWS CLI

Per aggiornare l'iscrizione a una sala

L'esempio seguente modifica il ruolo del membro della chat room specificato in `Administrator`.

```

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

```

Output:

```

{
  "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"
}
}

```

Per ulteriori informazioni, consulta [Creare una chat room](#) nella Guida per l'utente di Amazon Chime.

- Per API i dettagli, consulta [UpdateRoomMembership AWS CLI Command Reference](#).

## update-room

Il seguente esempio di codice mostra come utilizzare `update-room`.

### AWS CLI

Per aggiornare una chat room

L'esempio seguente modifica il nome della chat room specificata.

```

aws chime update-room \
  --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
  --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
  --name teamRoom

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Creare una chat room](#) nella Guida per l'utente di Amazon Chime.

- Per API i dettagli, consulta [UpdateRoom AWS CLI](#) Command Reference.

## update-user-settings

Il seguente esempio di codice mostra come utilizzare `update-user-settings`.

### AWS CLI

Per aggiornare le impostazioni utente

L'esempio seguente consente all'utente specificato di effettuare chiamate in entrata e in uscita e inviare e ricevere SMS messaggi.

```
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}"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Gestione dei numeri di telefono degli utenti](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta [UpdateUserSettings AWS CLI](#) Command Reference.

## update-user

Il seguente esempio di codice mostra come utilizzare `update-user`.

### AWS CLI

Per aggiornare i dettagli dell'utente

Questo esempio aggiorna i dettagli specificati per l'utente specificato.

Comando:

```
aws chime update-user \  
  --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \  
  --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE \  
  --user-name EXAMPLE
```

```
--license-type "Basic"
```

Output:

```
{
  "User": {
    "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
  }
}
```

- Per API i dettagli, vedere [UpdateUser](#) in AWS CLI Command Reference.

## update-voice-connector-group

Il seguente esempio di codice mostra come utilizzare `update-voice-connector-group`.

AWS CLI

Per aggiornare i dettagli di un gruppo Amazon Chime Voice Connector

L'update-voice-connector-groupesempio seguente aggiorna i dettagli del gruppo Amazon Chime Voice Connector specificato.

```
aws chime update-voice-connector-group \
  --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jk18901 \
  --name "newGroupName" \
  --voice-connector-items VoiceConnectorId=abcdef1ghij2klmno3pqr4,Priority=1
```

Output:

```
{
  "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"
  }
}
```

```
}  
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di Amazon Chime Voice Connector](#) nella Amazon Chime Administration Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateVoiceConnectorGroup](#)Reference.

## update-voice-connector

Il seguente esempio di codice mostra come utilizzare `update-voice-connector`.

### AWS CLI

Per aggiornare i dettagli di un Amazon Chime Voice Connector

L'`update-voice-connector` seguente aggiorna il nome del connettore Amazon Chime Voice specificato.

```
aws chime update-voice-connector \  
  --voice-connector-id abcdef1ghij2klmno3pqr4 \  
  --name newName \  
  --require-encryption
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Lavorare con i connettori vocali di Amazon Chime nella Guida](#) all'amministrazione di Amazon Chime.

- Per API i dettagli, consulta AWS CLI Command [UpdateVoiceConnector](#)Reference.



## API Esempi di Cloud Control utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface con Cloud Control API.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-resource**

Il seguente esempio di codice mostra come utilizzare `create-resource`.

AWS CLI

Per creare una risorsa

L'`create-resource` esempio seguente crea una risorsa AWS: :Kinesis: :Stream, denominata `ResourceExample`, con un periodo di conservazione di 168 ore e un numero di frammenti pari a tre.

```
aws cloudcontrol create-resource \  
  --type-name AWS::Kinesis::Stream \  
  --desired-state '{"Name": "ResourceExample", "RetentionPeriodHours":168, \  
  "ShardCount":3}'
```

Output:

```
{  
  "ProgressEvent": {  
    "EventTime": 1632506656.706,  
    "TypeName": "AWS::Kinesis::Stream",  
    "OperationStatus": "IN_PROGRESS",
```

```
    "Operation": "CREATE",
    "Identifier": "ResourceExample",
    "RequestToken": "20999d87-e304-4725-ad84-832dcbfd7fc5"
  }
}
```

Per ulteriori informazioni, consulta [Creating a resource](#) in the Cloud Control User Guide. API

- Per API i dettagli, consulta [CreateResource AWS CLI Command Reference](#).

## delete-resource

Il seguente esempio di codice mostra come utilizzare `delete-resource`.

### AWS CLI

Per eliminare una risorsa

L'esempio seguente elimina una risorsa `AWS::Kinesis::Stream` con l'identificatore `ResourceExample` dal tuo account. AWS

```
aws cloudcontrol delete-resource \
  --type-name AWS::Kinesis::Stream \
  --identifier ResourceExample
```

Output:

```
{
  "ProgressEvent": {
    "TypeName": "AWS::Kinesis::Stream",
    "Identifier": "ResourceExample",
    "RequestToken": "e48f26ff-d0f9-4ab8-a878-120db1edf111",
    "Operation": "DELETE",
    "OperationStatus": "IN_PROGRESS",
    "EventTime": 1632950300.14
  }
}
```

Per ulteriori informazioni, consulta [Eliminazione di una risorsa](#) nella Guida per l'utente di Cloud Control. API

- Per API i dettagli, consulta [DeleteResource AWS CLI Command Reference](#).

## get-resource-request-status

Il seguente esempio di codice mostra come utilizzare `get-resource-request-status`.

### AWS CLI

Per ottenere le informazioni sullo stato di una richiesta di risorse

L'esempio seguente restituisce informazioni sullo stato della richiesta di risorse specificata.

```
aws cloudcontrol get-resource-request-status \
  --request-token "e1a6b86e-46bd-41ac-bfba-001234567890"
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Gestire le richieste di funzionamento delle risorse](#) nella Guida per l'API utente di Cloud Control.

- Per API i dettagli, consulta [GetResourceRequestStatus AWS CLI Command Reference](#).

## get-resource

Il seguente esempio di codice mostra come utilizzare `get-resource`.

### AWS CLI

Per ottenere lo stato attuale di una risorsa

L'`get-resource` seguente restituisce lo stato corrente della risorsa AWS: :Kinesis: :Stream denominata. `ResourceExample`

```
aws cloudcontrol get-resource \
  --type-name AWS::Kinesis::Stream \
  --identifier ResourceExample
```

Output:

```
{
  "TypeName": "AWS::Kinesis::Stream",
  "ResourceDescription": {
    "Identifier": "ResourceExample",
    "Properties": "{\"Arn\":\"arn:aws:kinesis:us-west-2:099908667365:stream/ResourceExample\", \"RetentionPeriodHours\":168, \"Name\":\"ResourceExample\", \"ShardCount\":3}"
  }
}
```

Per ulteriori informazioni, consulta [Leggere lo stato attuale di una risorsa](#) nella Guida per l'utente di Cloud Control API.

- Per API i dettagli, consulta [GetResource AWS CLI Command Reference](#).

## list-resource-requests

Il seguente esempio di codice mostra come utilizzare `list-resource-requests`.

AWS CLI

Per elencare le richieste di funzionamento delle risorse attive

L'`list-resource-requests` seguente elenca le richieste di risorse CREATE e le UPDATE operazioni che non sono riuscite nell' AWS account.

```
aws cloudcontrol list-resource-requests \
  --resource-request-status-filter Operations=CREATE,OperationStatuses=FAILED
```

Output:

```
{
```

```

    "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Gestire le richieste di funzionamento delle risorse](#) nella Guida per l'API utente di Cloud Control.

- Per API i dettagli, consulta [ListResourceRequests AWS CLI Command Reference](#).

## list-resources

Il seguente esempio di codice mostra come utilizzare `list-resources`.

### AWS CLI

Per elencare le risorse di un determinato tipo

L'`list-resources` esempio seguente elenca le risorse AWS: `:Kinesis: :Stream` fornite nel tuo account. AWS

```

aws cloudcontrol list-resources \
  --type-name AWS::Kinesis::Stream

```

Output:

```

{
  "TypeName": "AWS::Kinesis::Stream",
  "ResourceDescriptions": [
    {
      "Identifier": "MyKinesisStream",
      "Properties": "{\"Name\": \"MyKinesisStream\"}"
    }
  ]
}

```

```

    },
    {
      "Identifier": "AnotherStream",
      "Properties": "{\"Name\": \"AnotherStream\"}"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Discovering resources](#) in the Cloud Control User Guide. API

- Per API i dettagli, consulta [ListResources AWS CLI Command Reference](#).

## update-resource

Il seguente esempio di codice mostra come utilizzare `update-resource`.

### AWS CLI

Per aggiornare le proprietà di una risorsa esistente

L'esempio seguente aggiorna la politica di conservazione di una LogGroup risorsa AWS: `:Logs:` denominata `ExampleLogGroup` a 90 giorni.

```

aws cloudcontrol update-resource \
  --type-name AWS::Logs::LogGroup \
  --identifier ExampleLogGroup \
  --patch-document "[{\"op\": \"replace\", \"path\": \"/RetentionInDays\", \"value\": 90}]"

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Aggiornamento di una risorsa nella Guida](#) per l'API utente di Cloud Control.

- Per API i dettagli, consulta [UpdateResource AWS CLI](#) Command Reference.

## AWS Cloud Map esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Cloud Map.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **create-private-dns-namespace**

Il seguente esempio di codice mostra come utilizzare `create-private-dns-namespace`.

#### AWS CLI

Per creare uno spazio dei DNS nomi privato

L'`create-private-dns-namespace` esempio seguente crea uno spazio dei nomi privato DNS.

```
aws servicediscovery create-private-dns-namespace \  
  --name example.com \  
  --vpc vpc-1c56417b
```

Output:

```
{  
  "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
```

```
}
```

Per confermare che l'operazione è riuscita, puoi eseguire `get-operation`. Per ulteriori informazioni, vedere [get-operation](#).

Per ulteriori informazioni, consulta [Creazione di namespace](#) nella AWS Cloud Map Developer Guide.

- Per API i dettagli, consulta Command [CreatePrivateDnsNamespace](#) Reference AWS CLI .

## create-service

Il seguente esempio di codice mostra come utilizzare `create-service`.

### AWS CLI

Per creare un servizio

L'esempio seguente crea un servizio.

```
aws servicediscovery create-service \  
  --name myservice \  
  --namespace-id ns-ylexjili4cdxy3xm \  
  --dns-config "NamespaceId=ns-ylexjili4cdxy3xm,RoutingPolicy=MULTIVALUE,DnsRecords=[{Type=A,TTL=60}]"
```

Output:

```
{  
  "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"
}
}

```

Per ulteriori informazioni, consulta [Creating services](#) in AWS Cloud Map Developer Guide.

- Per API i dettagli, consulta [CreateService AWS CLI](#) Command Reference.

## delete-namespace

Il seguente esempio di codice mostra come utilizzare `delete-namespace`.

### AWS CLI

Per eliminare un namespace

L'esempio seguente elimina uno spazio dei nomi.

```
aws servicediscovery delete-namespace \
  --id ns-ylexjili4cdxy3xm
```

Output:

```
{
  "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k98y6dtk"
}
```

Per confermare che l'operazione è riuscita, puoi eseguire `get-operation`. Per ulteriori informazioni, vedere [get-operation](#).

Per ulteriori informazioni, consulta [Eliminazione dei namespace nella AWS Cloud Map Developer Guide](#).

- Per i API dettagli, consulta [DeleteNamespace](#) Command Reference.AWS CLI

## delete-service

Il seguente esempio di codice mostra come utilizzare `delete-service`.

## AWS CLI

Per eliminare un servizio

L'`delete-service` seguente elimina un servizio.

```
aws servicediscovery delete-service \  
  --id srv-p5zdwlg5uvvzjita
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione dei servizi](#) nella AWS Cloud Map Developer Guide.

- Per API i dettagli, consulta [DeleteService AWS CLI Command Reference](#).

## `deregister-instance`

Il seguente esempio di codice mostra come utilizzare `deregister-instance`.

## AWS CLI

Per annullare la registrazione di un'istanza del servizio

L'`deregister-instance` seguente annulla la registrazione di un'istanza di servizio.

```
aws servicediscovery deregister-instance \  
  --service-id srv-p5zdwlg5uvvzjita \  
  --instance-id myservice-53
```

Output:

```
{  
  "OperationId": "4yejorelbukcjpnr6t1mrghsjwpngf4-k98rnaiq"  
}
```

Per confermare che l'operazione è riuscita, puoi eseguire `get-operation`. Per ulteriori informazioni, vedere [get-operation](#).

Per ulteriori informazioni, consulta [Annullamento della registrazione delle istanze di servizio](#) nella AWS Cloud Map Developer Guide.

- Per API i dettagli, consulta Command [DeregisterInstance](#)Reference AWS CLI .

## discover-instances

Il seguente esempio di codice mostra come utilizzare `discover-instances`.

### AWS CLI

Per scoprire le istanze registrate

L'`discover-instances`esempio seguente scopre le istanze registrate.

```
aws servicediscovery discover-instances \  
  --namespace-name example.com \  
  --service-name myservice \  
  --max-results 10 \  
  --health-status ALL
```

Output:

```
{  
  "Instances": [  
    {  
      "InstanceId": "myservice-53",  
      "NamespaceName": "example.com",  
      "ServiceName": "myservice",  
      "HealthStatus": "UNKNOWN",  
      "Attributes": {  
        "AWS_INSTANCE_IPV4": "172.2.1.3",  
        "AWS_INSTANCE_PORT": "808"  
      }  
    }  
  ]  
}
```

- Per API i dettagli, vedere [DiscoverInstances](#)in AWS CLI Command Reference.

## get-operation

Il seguente esempio di codice mostra come utilizzare `get-operation`.

## AWS CLI

Per ottenere il risultato di un'operazione

L'`get-operation`esempio seguente ottiene il risultato di un'operazione.

```
aws servicediscovery get-operation \  
  --operation-id gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd
```

Output:

```
{  
  "Operation": {  
    "Id": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd",  
    "Type": "CREATE_NAMESPACE",  
    "Status": "SUCCESS",  
    "CreateDate": 1587055860.121,  
    "UpdateDate": 1587055900.469,  
    "Targets": {  
      "NAMESPACE": "ns-ylexjili4cdxy3xm"  
    }  
  }  
}
```

- Per API i dettagli, vedere [GetOperation](#)in AWS CLI Command Reference.

## list-instances

Il seguente esempio di codice mostra come utilizzare `list-instances`.

### AWS CLI

Per elencare le istanze del servizio

L'`list-instances`esempio seguente elenca le istanze del servizio.

```
aws servicediscovery list-instances \  
  --service-id srv-qzpwvt2tfqcegapy
```

Output:

```
{
```

```
"Instances": [  
  {  
    "Id": "i-06bdabbae60f65a4e",  
    "Attributes": {  
      "AWS_INSTANCE_IPV4": "172.2.1.3",  
      "AWS_INSTANCE_PORT": "808"  
    }  
  }  
]
```

Per ulteriori informazioni, consulta [Visualizzazione di un elenco di istanze di servizio](#) nella AWS Cloud Map Developer Guide.

- Per API i dettagli, consulta [ListInstances AWS CLI](#) Command Reference.

## list-namespaces

Il seguente esempio di codice mostra come utilizzare `list-namespaces`.

### AWS CLI

Per elencare i namespace

L'`list-namespaces` esempio seguente elenca i namespace.

```
aws servicediscovery list-namespaces
```

Output:

```
{  
  "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-
pocfyjtrismwtvcxx",
    "CreateDate": 1586468974.698,
    "Description": "My second namespace",
    "Id": "ns-pocfyjtrismwtvcxx",
    "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"
}
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione di un elenco di namespace nella AWS Cloud Map Developer Guide](#).

- Per API i dettagli, consulta Command [ListNamespaces](#)Reference AWS CLI .

## list-services

Il seguente esempio di codice mostra come utilizzare `list-services`.

### AWS CLI

Per elencare i servizi

L'`list-services` seguente elenca i servizi.

```
aws servicediscovery list-services
```

Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta [Visualizzazione di un elenco di servizi](#) nella AWS Cloud Map Developer Guide.

- Per API i dettagli, consulta [ListServices AWS CLI Command Reference](#).

## register-instance

Il seguente esempio di codice mostra come utilizzare `register-instance`.

## AWS CLI

Per registrare un'istanza di servizio

L'`register-instance` seguente registra un'istanza di servizio.

```
aws servicediscovery register-instance \  
  --service-id srv-p5zdwlq5uvvzjita \  
  --instance-id myservice-53 \  
  --attributes=AWS_INSTANCE_IPV4=172.2.1.3,AWS_INSTANCE_PORT=808
```

Output:

```
{  
  "OperationId": "4yejorelbukcjzpnr6t1mrghsjwpngf4-k95yg2u7"  
}
```

Per confermare che l'operazione è riuscita, puoi eseguire `get-operation`. Per ulteriori informazioni, vedere [get-operation](#).

Per ulteriori informazioni, consulta [Registrazione delle istanze](#) nella AWS Cloud Map Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [RegisterInstance](#) Reference.

## AWS Cloud9 esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Cloud9.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)



## Azioni

### **create-environment-ec2**

Il seguente esempio di codice mostra come utilizzare `create-environment-ec2`.

#### AWS CLI

Per creare un ambiente di AWS sviluppo EC2 Cloud9

L'esempio seguente crea un AWS ambiente di sviluppo Cloud9 con le impostazioni specificate, avvia un'istanza Amazon Elastic Compute Cloud (EC2 Amazon) e quindi si connette dall'istanza all'ambiente.

```
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
```

Output:

```
{  
  "environmentId": "8a34f51ce1e04a08882f1e811bd706EX"  
}
```

Per ulteriori informazioni, consulta [Creazione di un EC2 ambiente nella Guida](#) per l'utente di AWS Cloud9.

- Per API i dettagli, consulta [CreateEnvironmentEc2](#) in AWS CLI Command Reference.

### **create-environment-membership**

Il seguente esempio di codice mostra come utilizzare `create-environment-membership`.

#### AWS CLI

Per aggiungere un membro dell'ambiente a un ambiente di sviluppo AWS Cloud9

Questo esempio aggiunge il membro dell'ambiente specificato all'ambiente di sviluppo AWS Cloud9 specificato.

Comando:

```
aws cloud9 create-environment-membership --environment-id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/AnotherDemoUser --permissions read-write
```

Output:

```
{
  "membership": {
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
    "userId": "AIDAJ3LOROMOUCTBSUGEX",
    "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
    "permissions": "read-write"
  }
}
```

- Per API i dettagli, consulta AWS CLI Command [CreateEnvironmentMembership](#)Reference.

## **delete-environment-membership**

Il seguente esempio di codice mostra come utilizzare `delete-environment-membership`.

AWS CLI

Per eliminare un membro dell'ambiente da un ambiente di sviluppo AWS Cloud9

Questo esempio elimina il membro dell'ambiente specificato dall'ambiente di sviluppo AWS Cloud9 specificato.

Comando:

```
aws cloud9 delete-environment-membership --environment-id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/AnotherDemoUser
```

Output:

```
None.
```

- Per API i dettagli, consulta Command [DeleteEnvironmentMembership](#)Reference AWS CLI .

## delete-environment

Il seguente esempio di codice mostra come utilizzare `delete-environment`.

### AWS CLI

Per eliminare un ambiente di AWS sviluppo Cloud9

Questo esempio elimina l'ambiente di sviluppo AWS Cloud9 specificato. Se un'EC2 istanza Amazon è connessa all'ambiente, interrompe anche l'istanza.

Comando:

```
aws cloud9 delete-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX
```

Output:

```
None .
```

- Per API i dettagli, consulta [DeleteEnvironment AWS CLI Command Reference](#).

## describe-environment-memberships

Il seguente esempio di codice mostra come utilizzare `describe-environment-memberships`.

### AWS CLI

Per ottenere informazioni sui membri dell'ambiente per un ambiente di sviluppo AWS Cloud9

Questo esempio ottiene informazioni sui membri dell'ambiente per l'ambiente di sviluppo AWS Cloud9 specificato.

Comando:

```
aws cloud9 describe-environment-memberships --environment-  
id 8a34f51ce1e04a08882f1e811bd706EX
```

Output:

```
{  
  "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"
}
]
}
```

Per ottenere informazioni sul proprietario di un ambiente di sviluppo AWS Cloud9

Questo esempio ottiene informazioni sul proprietario dell'ambiente di sviluppo AWS Cloud9 specificato.

Comando:

```
aws cloud9 describe-environment-memberships --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --permissions owner
```

Output:

```
{
"memberships": [
{
"environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
"userId": "AIDAJNUEDQAQWFELJDLEX",
"userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
"permissions": "owner"
}
]
}
```

Per ottenere informazioni su un membro dell'ambiente per più ambienti di sviluppo AWS Cloud9

Questo esempio ottiene informazioni sul membro dell'ambiente specificato per più ambienti di sviluppo AWS Cloud9.

Comando:

```
aws cloud9 describe-environment-memberships --user-arn arn:aws:iam::123456789012:user/MyDemoUser
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, consulta AWS CLI Command [DescribeEnvironmentMemberships](#) Reference.

## describe-environment-status

Il seguente esempio di codice mostra come utilizzare `describe-environment-status`.

### AWS CLI

Per ottenere informazioni sullo stato di un ambiente di sviluppo AWS Cloud9

Questo esempio ottiene informazioni sullo stato per l'ambiente di sviluppo AWS Cloud9 specificato.

Comando:

```
aws cloud9 describe-environment-status --environment-id 685f892f431b45c2b28cb69eadcdb0EX
```

**Output:**

```
{
  "status": "ready",
  "message": "Environment is ready to use"
}
```

- Per API i dettagli, consulta AWS CLI Command [DescribeEnvironmentStatus](#)Reference.

**describe-environments**

Il seguente esempio di codice mostra come utilizzare `describe-environments`.

**AWS CLI**

Per ottenere informazioni sugli ambienti di sviluppo AWS Cloud9

Questo esempio ottiene informazioni sugli ambienti di sviluppo AWS Cloud9 specificati.

Comando:

```
aws cloud9 describe-environments --environment-
ids 685f892f431b45c2b28cb69eadcdb0EX 349c86d4579e4e7298d500ff57a6b2EX
```

**Output:**

```
{
  "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"
    }
  }
]
```

- Per API i dettagli, consulta AWS CLI Command [DescribeEnvironments](#)Reference.

## list-environments

Il seguente esempio di codice mostra come utilizzare `list-environments`.

### AWS CLI

Per ottenere un elenco degli identificatori dell'ambiente di AWS sviluppo Cloud9 disponibili

Questo esempio ottiene un elenco di identificatori dell'ambiente di AWS sviluppo Cloud9 disponibili.

Comando:

```
aws cloud9 list-environments
```

Output:

```
{
  "environmentIds": [
    "685f892f431b45c2b28cb69eadcdb0EX",
    "1980b80e5f584920801c09086667f0EX"
  ]
}
```

- Per API i dettagli, consulta Command [ListEnvironments](#)Reference AWS CLI .

## update-environment-membership

Il seguente esempio di codice mostra come utilizzare `update-environment-membership`.

### AWS CLI

Per modificare le impostazioni di un membro dell'ambiente esistente per un ambiente di sviluppo AWS Cloud9

Questo esempio modifica le impostazioni del membro di ambiente esistente specificato per l'ambiente di sviluppo AWS Cloud9 specificato.

Comando:

```
aws cloud9 update-environment-membership --environment-id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/AnotherDemoUser --permissions read-only
```

Output:

```
{
  "membership": {
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
    "userId": "AIDAJ3LOROMOUCTBSU6EX",
    "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
    "permissions": "read-only"
  }
}
```

- Per API i dettagli, vedete [UpdateEnvironmentMembership](#) in AWS CLI Command Reference.

## update-environment

Il seguente esempio di codice mostra come utilizzare `update-environment`.

### AWS CLI

Per modificare le impostazioni di un ambiente di sviluppo AWS Cloud9 esistente

Questo esempio modifica le impostazioni specificate dell'ambiente di sviluppo AWS Cloud9 esistente specificato.

Comando:





```
--type RESOURCE \  
--type-name Example::Test::1234567890abcdef0 \  
--type-name-alias Example::Test::Alias
```

Output:

```
{  
  "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/Example-  
Test-Alias"  
}
```

Per ulteriori informazioni, consulta [Using the AWS CloudFormation registry](#) nella AWS CloudFormation User Guide.

- Per API i dettagli, vedere [ActivateType](#) in AWS CLI Command Reference.

## batch-describe-type-configurations

Il seguente esempio di codice mostra come utilizzare `batch-describe-type-configurations`.

AWS CLI

Per descrivere in batch una configurazione di tipo

L'`batch-describe-type-configurations` seguente configura i dati per il tipo.

```
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"
```

Output:

```
{  
  "Errors": [],  
  "UnprocessedTypeConfigurations": [],  
  "TypeConfigurations": [  
    {  
      "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/  
Example-Test-Type",  
      "Alias": "MyConfiguration",
```

```

    "Configuration": "{\n
    \n      \"Example\": {\n
    \n        \"ApiKey\":
    \n        \"examplekey\", \n
    \n        \"ApplicationKey\": \"examplekey1\", \n
    \n        \"ApiURL\": \"exampleurl\" \n
    \n      }\n
    \n    },
    \n    \"LastUpdated\": \"2021-10-01T15:25:46.210000+00:00\",
    \n    \"TypeArn\": \"arn:aws:cloudformation:us-east-1:123456789012:type/resource/
    Example-Test-Type\"
  }
]
}

```

Per ulteriori informazioni, vedere [Utilizzo del AWS CloudFormation registro](#) nella Guida per l'AWS CloudFormation utente.

- Per API i dettagli, vedere [BatchDescribeTypeConfigurations](#) in AWS CLI Command Reference.

## cancel-update-stack

Il seguente esempio di codice mostra come utilizzare `cancel-update-stack`.

### AWS CLI

Per annullare un aggiornamento dello stack in corso

Il `cancel-update-stack` comando seguente annulla un aggiornamento dello stack nello stack: `myteststack`

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- Per API i dettagli, vedere [CancelUpdateStack](#) in AWS CLI Command Reference.

## continue-update-rollback

Il seguente esempio di codice mostra come utilizzare `continue-update-rollback`.

### AWS CLI

Per riprovare un rollback di aggiornamento

L'`continue-update-rollback` esempio seguente riprende un'operazione di rollback da un aggiornamento dello stack precedentemente non riuscito.

```
aws cloudformation continue-update-rollback \
```

```
--stack-name my-stack
```

Questo comando non produce alcun output.

- Per i API dettagli, vedere [ContinueUpdateRollback](#) in Command Reference.AWS CLI

## create-change-set

Il seguente esempio di codice mostra come utilizzare `create-change-set`.

### AWS CLI

Per creare un set di modifiche

L'`create-change-set` esempio seguente crea un set di modifiche con la `CAPABILITY_IAM` funzionalità. Il file `template.yaml` è un AWS CloudFormation modello nella cartella corrente che definisce uno stack che include IAM risorse.

```
aws cloudformation create-change-set \  
  --stack-name my-application \  
  --change-set-name my-change-set \  
  --template-body file://template.yaml \  
  --capabilities CAPABILITY_IAM
```

Output:

```
{  
  "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"  
}
```

- Per API i dettagli, vedere [CreateChangeSet](#) in AWS CLI Command Reference.

## create-stack-instances

Il seguente esempio di codice mostra come utilizzare `create-stack-instances`.

### AWS CLI

Per creare istanze stack

L'`create-stack-instances` seguente crea istanze di uno stack impostato in due account e in quattro regioni. L'impostazione della tolleranza agli errori garantisce che l'aggiornamento venga tentato in tutti gli account e le regioni, anche se non è possibile creare alcuni stack.

```
aws cloudformation create-stack-instances \  
  --stack-set-name my-stack-set \  
  --accounts 123456789012 223456789012 \  
  --regions us-east-1 us-east-2 us-west-1 us-west-2 \  
  --operation-preferences FailureToleranceCount=7
```

Output:

```
{  
  "OperationId": "d7995c31-83c2-xmpl-a3d4-e9ca2811563f"  
}
```

Per creare un set di stack, usa il comando `create-stack-set`

- Per API i dettagli, vedere [CreateStackInstances](#) in AWS CLI Command Reference.

## `create-stack-set`

Il seguente esempio di codice mostra come utilizzare `create-stack-set`.

### AWS CLI

Per creare un set di stack

L'`create-stack-set` seguente crea un set di stack utilizzando il modello di YAML file specificato. `template.yaml` è un AWS CloudFormation modello nella cartella corrente che definisce uno stack.

```
aws cloudformation create-stack-set \  
  --stack-set-name my-stack-set \  
  --template-body file://template.yaml \  
  --description "SNS topic"
```

Output:

```
{
```

```
"StackSetId": "my-stack-set:8d0f160b-d157-xmpl-a8e6-c0ce8e5d8cc1"  
}
```

Per aggiungere istanze dello stack allo stack set, utilizzate il comando `create-stack-instances`

- Per API i dettagli, consulta Command [CreateStackSet](#) Reference AWS CLI .

## create-stack

Il seguente esempio di codice mostra come utilizzare `create-stack`.

### AWS CLI

Per creare una AWS CloudFormation pila

Il `create-stacks` comando seguente crea uno stack con il nome `myteststack` utilizzando il `sampletemplate.json` modello:

```
aws cloudformation create-stack --stack-name myteststack --template-body file://sampletemplate.json --parameters ParameterKey=KeyPairName,ParameterValue=TestKey  
ParameterKey=SubnetIDs,ParameterValue=SubnetID1\\,SubnetID2
```

Output:

```
{  
  "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/  
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"  
}
```

Per ulteriori informazioni, consulta [Stacks](#) nella Guida per l'AWS CloudFormation utente.

- Per API i dettagli, vedere [CreateStack](#) in AWS CLI Command Reference.

## deactivate-type

Il seguente esempio di codice mostra come utilizzare `deactivate-type`.

### AWS CLI

Per disattivare un tipo

L'`deactivate-type`esempio seguente disattiva un'estensione pubblica precedentemente attivata in questo account e nella regione.

```
aws cloudformation deactivate-type \  
  --region us-west-2 \  
  --type MODULE \  
  --type-name Example::Test::Type::MODULE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Utilizzo del AWS CloudFormation registro nella Guida](#) per l'AWS CloudFormation utente.

- Per API i dettagli, vedere [DeactivateType](#)in AWS CLI Command Reference.

## `delete-change-set`

Il seguente esempio di codice mostra come utilizzare `delete-change-set`.

### AWS CLI

Per eliminare un set di modifiche

L'`delete-change-set`esempio seguente elimina un set di modifiche specificando il nome del set di modifiche e il nome dello stack.

```
aws cloudformation delete-change-set \  
  --stack-name my-stack \  
  --change-set-name my-change-set
```

Questo comando non produce alcun output.

L'`delete-change-set`esempio seguente elimina un set di modifiche specificando l'intero ARN set di modifiche.

```
aws cloudformation delete-change-set \  
  --change-set-name arn:aws:cloudformation:us-east-2:123456789012:changeSet/my-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteChangeSet](#)in AWS CLI Command Reference.

## delete-stack-instances

Il seguente esempio di codice mostra come utilizzare `delete-stack-instances`.

### AWS CLI

Per eliminare le istanze dello stack

L'esempio seguente elimina le istanze di uno stack impostato in due account in due regioni e termina gli stack.

```
aws cloudformation delete-stack-instances \
  --stack-set-name my-stack-set \
  --accounts 123456789012 567890123456 \
  --regions us-east-1 us-west-1 \
  --no-retain-stacks
```

Output:

```
{
  "OperationId": "ad49f10c-fd1d-413f-a20a-8de6e2fa8f27"
}
```

Per eliminare un set di stack vuoto, utilizzate il comando `delete-stack-set`

- Per API i dettagli, vedere [DeleteStackInstances](#) in AWS CLI Command Reference.

## delete-stack-set

Il seguente esempio di codice mostra come utilizzare `delete-stack-set`.

### AWS CLI

Per eliminare uno stack set

Il comando seguente elimina il set di stack vuoto specificato. Il set di stack deve essere vuoto.

```
aws cloudformation delete-stack-set \
  --stack-set-name my-stack-set
```

Questo comando non produce alcun output.

Per eliminare le istanze dallo stack set, utilizzate il comando `delete-stack-instances`



- Per API i dettagli, consultate AWS CLI Command [DeleteStackSet](#) Reference.

## delete-stack

Il seguente esempio di codice mostra come utilizzare `delete-stack`.

### AWS CLI

Per eliminare una pila

L'`delete-stack` esempio seguente elimina lo stack specificato.

```
aws cloudformation delete-stack \  
  --stack-name my-stack
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteStack](#) in AWS CLI Command Reference.

## deploy

Il seguente esempio di codice mostra come utilizzare `deploy`.

### AWS CLI

Il comando seguente distribuisce il modello denominato `template.json` in uno stack denominato: `my-new-stack`

```
aws cloudformation deploy --template-file /path_to_template/template.json \  
  --stack-name my-new-stack --parameter-overrides Key1=Value1 Key2=Value2 -- \  
  tags Key1=Value1 Key2=Value2
```

- Per API i dettagli, consulta [Deploy](#) in AWS CLI Command Reference.

## deregister-type

Il seguente esempio di codice mostra come utilizzare `deregister-type`.

### AWS CLI

Per annullare la registrazione di una versione del tipo

L'operazione `deregister-type` seguente rimuove la versione del tipo specificato dall'uso attivo nel CloudFormation registro, in modo che non possa più essere utilizzata nelle CloudFormation operazioni.

```
aws cloudformation deregister-type \  
  --type RESOURCE \  
  --type-name My::Logs::LogGroup \  
  --version-id 00000002
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Using the CloudFormation Registry](#) in the AWS CloudFormation Users Guide.

- Per API i dettagli, vedere [DeregisterType](#) in AWS CLI Command Reference.

## **describe-account-limits**

Il seguente esempio di codice mostra come utilizzare `describe-account-limits`.

### AWS CLI

Per ottenere informazioni sui limiti del tuo account

Il comando seguente recupera un elenco di limiti regionali per l'account corrente.

```
aws cloudformation describe-account-limits
```

Output:

```
{  
  "AccountLimits": [  
    {  
      "Name": "StackLimit",  
      "Value": 200  
    },  
    {  
      "Name": "StackOutputsLimit",  
      "Value": 60  
    },  
    {
```

```

        "Name": "ConcurrentResourcesLimit",
        "Value": 2500
    }
]
}

```

- Per API i dettagli, vedere [DescribeAccountLimits](#) in AWS CLI Command Reference.

## describe-change-set

Il seguente esempio di codice mostra come utilizzare `describe-change-set`.

### AWS CLI

Per ottenere informazioni su un set di modifiche

L'`describe-change-set` esempio seguente visualizza i dettagli del set di modifiche specificato dal nome del set di modifiche e dal nome dello stack.

```

aws cloudformation describe-change-set \
  --change-set-name my-change-set \
  --stack-name my-stack

```

L'`describe-change-set` esempio seguente visualizza i dettagli del set di modifiche specificato dalla parte completa ARN del set di modifiche:

```

aws cloudformation describe-change-set \
  --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784

```

Output:

```

{
  "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
}

```

- Per API i dettagli, vedere [DescribeChangeSet](#) in AWS CLI Command Reference.

## describe-publisher

Il seguente esempio di codice mostra come utilizzare `describe-publisher`.

## AWS CLI

Per descrivere un editore

L'`describe-publisher` esempio seguente configura le informazioni per un editore.

```
aws cloudformation describe-publisher \  
  --region us-west-2 \  
  --publisher-id 000q6TfUovXsEMmgKowxDZLLwqr2QUsh
```

Output:

```
{  
  "PublisherId": "000q6TfUovXsEMmgKowxDZLLwqr2QUshd2e75c8c",  
  "PublisherStatus": "VERIFIED",  
  "IdentityProvider": "AWS_Marketplace",  
  "PublisherProfile": "https://aws.amazon.com/marketplace/seller-profile?  
id=2c5dc1f0-17cd-4259-8e46-822a83gdtegd"  
}
```

Per ulteriori informazioni, vedere [Using the AWS CloudFormation registry](#) in the AWS CloudFormation User Guide.

- Per API i dettagli, vedere [DescribePublisher](#) in AWS CLI Command Reference.

## `describe-stack-drift-detection-status`

Il seguente esempio di codice mostra come utilizzare `describe-stack-drift-detection-status`.

## AWS CLI

Per controllare lo stato di un'operazione di rilevamento della deriva

L'`describe-stack-drift-detection-status` esempio seguente mostra lo stato di un'operazione di rilevamento della deriva. Ottieni il by ID eseguendo il `detect-stack-drift` comando.

```
aws cloudformation describe-stack-drift-detection-status \  
  --stack-drift-detection-id 1a229160-e4d9-xmpl-ab67-0a4f93df83d4
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [DescribeStackDriftDetectionStatus](#) in AWS CLI Command Reference.

## describe-stack-events

Il seguente esempio di codice mostra come utilizzare `describe-stack-events`.

### AWS CLI

Per descrivere gli eventi dello stack

L'esempio seguente visualizza i 2 eventi più recenti per lo stack specificato.

```
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": "eyJ0ZXh0VG9XMPLiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}

```

- Per API i dettagli, vedere [DescribeStackEvents](#) in AWS CLI Command Reference.

## describe-stack-instance

Il seguente esempio di codice mostra come utilizzare `describe-stack-instance`.

### AWS CLI

Per descrivere un'istanza stack

Il comando seguente descrive un'istanza dello stack specificato impostato nell'account e nella regione specificati. Lo stack set si trova nella regione e nell'account correnti e l'istanza si trova nella `us-west-2` regione dell'account. `123456789012` :

```

aws cloudformation describe-stack-instance \
  --stack-set-name my-stack-set \
  --stack-instance-account 123456789012 \
  --stack-instance-region us-west-2

```

Output:

```

{
  "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,
ResourceType: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/
Fdt89BXMPLyiqbMNLjVk55Yqqvi3NYW2nKLUVWhUGEhNfCmZdyj967lhriaG/dWMobS040o=)."
  }
}

```

- Per API i dettagli, vedere [DescribeStackInstance](#) in AWS CLI Command Reference.

## describe-stack-resource-drifts

Il seguente esempio di codice mostra come utilizzare `describe-stack-resource-drifts`.

### AWS CLI

Per ottenere informazioni sulle risorse derivate dalla definizione dello stack

Il comando seguente visualizza informazioni sulle risorse alla deriva per lo stack specificato. Per avviare il rilevamento della deriva, utilizzate il comando `detect-stack-drift`:

```

aws cloudformation describe-stack-resource-drifts \
  --stack-name my-stack

```

L'output mostra una funzione AWS Lambda che è stata modificata: `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.\",
\nEnvironment\": {\"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\": \"2019-10-02T05:54:44.064Z\"
}
]
}
}

```

- Per API i dettagli, consulta [DescribeStackResourceDrifts AWS CLI Command Reference](#).

## describe-stack-resource

Il seguente esempio di codice mostra come utilizzare `describe-stack-resource`.

### AWS CLI

Per ottenere informazioni su una risorsa dello stack

L'esempio seguente visualizza i dettagli della risorsa denominata `MyFunction` nello stack specificato.

```
aws cloudformation describe-stack-resource \
```

```
--stack-name MyStack \  
--logical-resource-id MyFunction
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

- Per API i dettagli, vedere [DescribeStackResource](#) in AWS CLI Command Reference.

## describe-stack-resources

Il seguente esempio di codice mostra come utilizzare `describe-stack-resources`.

AWS CLI

Per ottenere informazioni su una risorsa dello stack

L'esempio seguente visualizza i dettagli delle risorse nello stack specificato.

```
aws cloudformation describe-stack-resources \  
--stack-name my-stack
```

Output:

```
{
```

```
"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"  
    }  
  }  
]
```

- Per API i dettagli, vedere [DescribeStackResources](#) in AWS CLI Command Reference.

## describe-stack-set-operation

Il seguente esempio di codice mostra come utilizzare `describe-stack-set-operation`.

### AWS CLI

Per ottenere informazioni su un'operazione di stack set

Il seguente `describe-stack-set-operation` `esempio mostra i dettagli di un'operazione di aggiornamento sul set di stack specificato.

```
aws cloudformation describe-stack-set-operation \  
  --stack-set-name enable-config \  
  --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0
```

Output:

```
{  
  "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"  
  }  
}
```

- Per API i dettagli, vedere [DescribeStackSetOperation](#) in AWS CLI Command Reference.

## describe-stack-set

Il seguente esempio di codice mostra come utilizzare `describe-stack-set`.

### AWS CLI

Per ottenere informazioni su uno stack set

Il seguente `describe-stack-set` esempio mostra i dettagli sul set di stack specificato.

```
aws cloudformation describe-stack-set \  
  --stack-set-name my-stack-set
```

Output:

```
{  
  "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"  
  }  
}
```

- Per API i dettagli, vedere [DescribeStackSet](#) in AWS CLI Command Reference.

## describe-stacks

Il seguente esempio di codice mostra come utilizzare `describe-stacks`.

### AWS CLI

Per descrivere le AWS CloudFormation pile

Il `describe-stacks` comando seguente mostra informazioni di riepilogo per lo `myteststack` stack:

```
aws cloudformation describe-stacks --stack-name myteststack
```

Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta `Stacks` nella Guida per l'AWS CloudFormation utente.

- Per API i dettagli, vedere [DescribeStacks](#) in AWS CLI Command Reference.

## describe-type-registration

Il seguente esempio di codice mostra come utilizzare `describe-type-registration`.

## AWS CLI

Per visualizzare le informazioni sulla registrazione dei tipi

L'`describe-type-registration` seguente visualizza informazioni sulla registrazione del tipo specificato, inclusi lo stato, il tipo e la versione corrente del tipo.

```
aws cloudformation describe-type-registration \  
  --registration-token a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Using the CloudFormation Registry](#) nella AWS CloudFormation Users Guide.

- Per API i dettagli, vedere [DescribeTypeRegistration](#) in AWS CLI Command Reference.

## `describe-type`

Il seguente esempio di codice mostra come utilizzare `describe-type`.

## AWS CLI

Per visualizzare le informazioni sul tipo

L'`describe-type` seguente visualizza le informazioni per il tipo specificato.

```
aws cloudformation describe-type \  
  --type-name My::Logs::LogGroup \  
  --type RESOURCE
```

Output:

```
{
  "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]"
}
```

Per ulteriori informazioni, vedere [Using the CloudFormation Registry](#) in the AWS CloudFormation Users Guide.

- Per API i dettagli, vedere [DescribeType](#) in AWS CLI Command Reference.

## detect-stack-drift

Il seguente esempio di codice mostra come utilizzare `detect-stack-drift`.

### AWS CLI

Per rilevare risorse alla deriva

L'esempio seguente avvia il rilevamento della deriva per lo stack specificato.

```
aws cloudformation detect-stack-drift \
  --stack-name my-stack
```

Output:

```
{
  "StackDriftDetectionId": "1a229160-e4d9-xmpl-ab67-0a4f93df83d4"
}
```



È quindi possibile utilizzare questo ID con il `describe-stack-resource-drifts` comando per descrivere le risorse alla deriva.

- Per API i dettagli, vedere [DetectStackDrift](#) in AWS CLI Command Reference.

## detect-stack-resource-drift

Il seguente esempio di codice mostra come utilizzare `detect-stack-resource-drift`.

### AWS CLI

Per rilevare la deriva di una risorsa

L'esempio seguente controlla una risorsa denominata `MyFunction` in uno stack denominato `driftMyStack`:

```
aws cloudformation detect-stack-resource-drift \
  --stack-name MyStack \
  --logical-resource-id MyFunction
```

L'output mostra una funzione AWS Lambda che è stata modificata: `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"
}

```

- Per API i dettagli, consulta [DetectStackResourceDrift AWS CLI Command Reference](#).

## detect-stack-set-drift

Il seguente esempio di codice mostra come utilizzare `detect-stack-set-drift`.

### AWS CLI

Per rilevare la deriva su un set di stack e su tutte le istanze di stack associate

L'`detect-stack-set-drift` esempio seguente avvia le operazioni di rilevamento della deriva sullo stack set specificato, incluse tutte le istanze di stack associate a tale set di stack, e restituisce un ID di operazione che può essere utilizzato per tenere traccia dello stato dell'operazione di drift.

```

aws cloudformation detect-stack-set-drift \
  --stack-set-name stack-set-drift-example

```

Output:

```

{
  "OperationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}

```

Per ulteriori informazioni, vedere [Rilevamento delle modifiche di configurazione non gestite](#) negli stack set nella Guida per l'utente.AWS CloudFormation

- Per API i dettagli, vedere [DetectStackSetDrift](#) in AWS CLI Command Reference.

## estimate-template-cost

Il seguente esempio di codice mostra come utilizzare `estimate-template-cost`.

### AWS CLI

Per stimare il costo del modello

L'esempio seguente genera una stima dei costi per un modello denominato `template.yaml` nella cartella corrente.

```
aws cloudformation estimate-template-cost \  
  --template-body file://template.yaml
```

Output:

```
{  
  "Url": "http://calculator.s3.amazonaws.com/calc5.html?  
key=cloudformation/7870825a-xmpl-4def-92e7-c4f8dd360cca"  
}
```

- Per API i dettagli, vedere [EstimateTemplateCost](#) in AWS CLI Command Reference.

## execute-change-set

Il seguente esempio di codice mostra come utilizzare `execute-change-set`.

### AWS CLI

Per eseguire un set di modifiche

L'esempio seguente esegue un set di modifiche specificato dal nome del set di modifiche e dal nome dello stack.

```
aws cloudformation execute-change-set \  
  --change-set-name my-change-set \  
  --stack-name my-stack
```

```
--stack-name my-stack
```

L'esecuzione seguente esegue un set di modifiche specificato dalla parte completa del set ARN di modifiche.

```
aws cloudformation execute-change-set \
  --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-
  change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784
```

- Per API i dettagli, vedere [ExecuteChangeSet](#) in AWS CLI Command Reference.

## get-stack-policy

Il seguente esempio di codice mostra come utilizzare `get-stack-policy`.

### AWS CLI

Per visualizzare una politica dello stack

L'esecuzione seguente mostra la politica dello stack per lo stack specificato. Per allegare una policy a uno stack, utilizzare il comando `set-stack-policy`

```
aws cloudformation get-stack-policy \
  --stack-name my-stack
```

Output:

```
{
  "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}"
```

- Per API i dettagli, vedere [GetStackPolicy](#) in AWS CLI Command Reference.

## get-template-summary

Il seguente esempio di codice mostra come utilizzare `get-template-summary`.

## AWS CLI

Per visualizzare un riepilogo del modello

Il comando seguente visualizza informazioni di riepilogo sulle risorse e i metadati per il file modello specificato.

```
aws cloudformation get-template-summary \  
--template-body file://template.yaml
```

Output:

```
{  
  "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"  
}
```

- Per API i dettagli, vedere [GetTemplateSummary](#) in AWS CLI Command Reference.

## get-template

Il seguente esempio di codice mostra come utilizzare `get-template`.

### AWS CLI

Per visualizzare il corpo del modello per uno AWS CloudFormation stack

Il `get-template` comando seguente mostra il modello per lo `myteststack` stack:

```
aws cloudformation get-template --stack-name myteststack
```

**Output:**

```
{
  "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"
        }
      }
    }
  }
}
```

- Per API i dettagli, vedere [GetTemplate](#) in AWS CLI Command Reference.

**list-change-sets**

Il seguente esempio di codice mostra come utilizzare `list-change-sets`.

**AWS CLI**

Per elencare i set di modifiche

L'`list-change-sets` esempio seguente visualizza un elenco dei set di modifiche in sospeso per lo stack specificato.

```
aws cloudformation list-change-sets \
```

```
--stack-name my-stack
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ListChangeSets](#) in AWS CLI Command Reference.

## list-exports

Il seguente esempio di codice mostra come utilizzare `list-exports`.

AWS CLI

Per elencare le esportazioni

L'`list-exports` esempio seguente visualizza un elenco delle esportazioni dagli stack nella regione corrente.

```
aws cloudformation list-exports
```

Output:

```
{
  "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-075ed3xmpllebd2fb1"
    },
    {
        "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
        "Name": "private-vpc-vpcid",
        "Value": "vpc-011d7xmpl1100e9841"
    }
]
}

```

- Per API i dettagli, vedere [ListExports](#) in AWS CLI Command Reference.

## list-imports

Il seguente esempio di codice mostra come utilizzare `list-imports`.

### AWS CLI

Per elencare le importazioni

L'`list-imports` esempio seguente elenca gli stack che importano l'esportazione specificata. Per ottenere l'elenco delle esportazioni disponibili, utilizzate il `list-exports` comando.

```

aws cloudformation list-imports \
  --export-name private-vpc-vpcid

```

Output:

```

{
  "Imports": [
    "my-database-stack"
  ]
}

```



```
]
}
```

- Per API i dettagli, vedere [ListImports](#) in AWS CLI Command Reference.

## list-stack-instances

Il seguente esempio di codice mostra come utilizzare `list-stack-instances`.

### AWS CLI

Per elencare le istanze di uno stack

L'`list-stack-instances` esempio seguente elenca le istanze create dal set di stack specificato.

```
aws cloudformation list-stack-instances \
  --stack-set-name enable-config
```

L'output di esempio include dettagli su uno stack che non è stato aggiornato a causa di un errore:

```
{
  "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)."
```

- Per API i dettagli, vedere [ListStackInstances](#) in AWS CLI Command Reference.

## list-stack-resources

Il seguente esempio di codice mostra come utilizzare `list-stack-resources`.

### AWS CLI

Per elencare le risorse in una pila

Il comando seguente visualizza l'elenco delle risorse nello stack specificato.

```
aws cloudformation list-stack-resources \
  --stack-name my-stack
```

Output:

```
{
  "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"
    }
  }
]
}

```

- Per API i dettagli, vedere [ListStackResources](#) in AWS CLI Command Reference.

## list-stack-set-operation-results

Il seguente esempio di codice mostra come utilizzare `list-stack-set-operation-results`.

### AWS CLI

Per elencare i risultati dell'operazione Stack Set

Il comando seguente visualizza i risultati di un'operazione di aggiornamento sulle istanze del set di stack specificato.

```

aws cloudformation list-stack-set-operation-results \
  --stack-set-name enable-config \
  --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0

```

Output:

```

{
  "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"
  }
]
}

```

Nota: lo SKIPPED stato di AccountGateResult è previsto per le operazioni riuscite a meno che non si crei una funzione account gate.

- Per API i dettagli, vedere [ListStackSetOperationResults](#) in AWS CLI Command Reference.

## list-stack-set-operations

Il seguente esempio di codice mostra come utilizzare `list-stack-set-operations`.

### AWS CLI

Per elencare le operazioni relative agli stack set

L'`list-stack-set-operations` esempio seguente visualizza l'elenco delle operazioni più recenti sullo stack set specificato.

```

aws cloudformation list-stack-set-operations \
  --stack-set-name my-stack-set

```

Output:

```

{
  "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"
  }
]
}
```

- Per API i dettagli, vedere [ListStackSetOperations](#) in AWS CLI Command Reference.

## list-stack-sets

Il seguente esempio di codice mostra come utilizzare `list-stack-sets`.

### AWS CLI

Per elencare i set di stack

L'`list-stack-sets` esempio seguente visualizza l'elenco dei set di stack nella regione e nell'account correnti.

```
aws cloudformation list-stack-sets
```

Output:

```
{
  "Summaries": [
    {
      "StackSetName": "enable-config",
      "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
      "Description": "Enable AWS Config",
      "Status": "ACTIVE"
    }
  ]
}
```

```
]
}
```

- Per API i dettagli, vedere [ListStackSets](#) in AWS CLI Command Reference.

## list-stacks

Il seguente esempio di codice mostra come utilizzare `list-stacks`.

### AWS CLI

Per elencare le AWS CloudFormation pile

Il `list-stacks` comando seguente mostra un riepilogo di tutti gli stack con uno stato di `CREATE_COMPLETE`

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

Output:

```
[
  {
    "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"
  }
]
```

- Per API i dettagli, vedere [ListStacks](#) in AWS CLI Command Reference.

## list-type-registrations

Il seguente esempio di codice mostra come utilizzare `list-type-registrations`.

## AWS CLI

Per elencare le registrazioni completate di un tipo

L'`list-type-registrations` seguente visualizza un elenco delle registrazioni di tipo completate per il tipo specificato.

```
aws cloudformation list-type-registrations \  
  --type RESOURCE \  
  --type-name My::Logs::LogGroup \  
  --registration-status-filter COMPLETE
```

Output:

```
{  
  "RegistrationTokenList": [  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
    "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"  
  ]  
}
```

Per ulteriori informazioni, vedere [Using the CloudFormation Registry](#) nella AWS CloudFormation Users Guide.

- Per API i dettagli, vedere [ListTypeRegistrations](#) in AWS CLI Command Reference.

## list-type-versions

Il seguente esempio di codice mostra come utilizzare `list-type-versions`.

### AWS CLI

Per elencare la versione di un'estensione

L'`list-type-versions` seguente restituisce informazioni di riepilogo sulle versioni di un'estensione.

```
aws cloudformation list-type-versions \  
  --endpoint https://example.com \  
  --region us-west-2 \  
  --type-name My::Logs::LogGroup
```

```
--type RESOURCE \  
--type-name My::Resource::Example \  
--publisher-id 123456789012
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Uso del AWS CloudFormation registro](#) nella Guida per l'AWS CloudFormation utente.

- Per API i dettagli, vedere [ListTypeVersions](#) in AWS CLI Command Reference.

## list-types

Il seguente esempio di codice mostra come utilizzare `list-types`.

### AWS CLI

Per elencare i tipi di risorse private presenti in un account

L'`list-types` esempio seguente visualizza un elenco dei tipi di risorse private attualmente registrati nell' AWS account corrente.

```
aws cloudformation list-types
```

Output:

```
{  
  "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"
    }
]
}
```

Per ulteriori informazioni, vedere [Using the CloudFormation Registry](#) nella AWS CloudFormation Users Guide.

- Per API i dettagli, vedere [ListTypes](#) in AWS CLI Command Reference.

## package

Il seguente esempio di codice mostra come utilizzare `package`.

### AWS CLI

Il comando seguente esporta un modello denominato `template.json` caricando artefatti locali nel bucket S3 `bucket-name` e scrive il modello esportato 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
```

- Per API i dettagli, vedere [Package](#) in AWS CLI Command Reference.

## publish-type

Il seguente esempio di codice mostra come utilizzare `publish-type`.

### AWS CLI

Per pubblicare un'estensione

L'esempio seguente pubblica l'estensione specificata nel CloudFormation registro come estensione pubblica in questa regione.

```
aws cloudformation publish-type \  
  --region us-west-2 \  
  --type RESOURCE \  
  --template template.json \  
  --output-template-file packaged-template.json \  
  --use-json
```

```
--type-name Example::Test::1234567890abcdef0
```

Output:

```
{
  "PublicTypeArn": "arn:aws:cloudformation:us-west-2::type/
resource/000q6TfUovXsEMmgKowxDZLlWqr2QUshd2e75c8c/Example-
Test-1234567890abcdef0/1.0.0"
}
```

Per ulteriori informazioni, vedere [Using the AWS CloudFormation registry](#) in the AWS CloudFormation User Guide.

- Per API i dettagli, vedere [PublishType](#) in AWS CLI Command Reference.

## register-publisher

Il seguente esempio di codice mostra come utilizzare `register-publisher`.

AWS CLI

Per registrare un editore

L'`register-publisher` esempio seguente registra un editore e accetta il parametro `terms and condition`.

```
aws cloudformation register-publisher \
  --region us-west-2 \
  --accept-terms-and-conditions
```

Output:

```
{
  "PublisherId": "000q6TfUovXsEMmgKowxDZLlWqr2QUshd2e75c8c"
}
```

Per ulteriori informazioni, vedere [Using the AWS CloudFormation registry](#) in the AWS CloudFormation User Guide.

- Per API i dettagli, vedere [RegisterPublisher](#) in AWS CLI Command Reference.

## register-type

Il seguente esempio di codice mostra come utilizzare `register-type`.

### AWS CLI

Per registrare un tipo di risorsa

L'esempio seguente registra il tipo di risorsa specificato come tipo di risorsa privata nell'account dell'utente.

```
aws cloudformation register-type \  
  --type-name My::Organization::ResourceName \  
  --schema-handler-package s3://bucket_name/my-organization-resource_name.zip \  
  --type RESOURCE
```

Output:

```
{  
  "RegistrationToken": "f5525280-104e-4d35-bef5-8f1f1example"  
}
```

Per ulteriori informazioni, vedere [Registrazione dei provider di risorse](#) nella Guida utente dell'interfaccia a riga di CloudFormation comando per lo sviluppo dei tipi.

- Per API i dettagli, vedere [RegisterType](#) in AWS CLI Command Reference.

## set-stack-policy

Il seguente esempio di codice mostra come utilizzare `set-stack-policy`.

### AWS CLI

Per applicare una politica di stack

L'esempio seguente disabilita gli aggiornamenti per la risorsa specificata nello stack specificato. `stack-policy.json` è un JSON documento che definisce le operazioni consentite sulle risorse dello stack.

```
aws cloudformation set-stack-policy \  
  --stack-name stack-name \  
  --policy-name stack-policy-name \  
  --policy-body stack-policy-body
```

```
--stack-name my-stack \  
--stack-policy-body file://stack-policy.json
```

Output:

```
{  
  "Statement" : [  
    {  
      "Effect" : "Allow",  
      "Action" : "Update:*",  
      "Principal": "*",  
      "Resource" : "*"   
    },  
    {  
      "Effect" : "Deny",  
      "Action" : "Update:*",  
      "Principal": "*",  
      "Resource" : "LogicalResourceId/bucket"   
    }  
  ]  
}
```

- Per API i dettagli, vedere [SetStackPolicy](#) in AWS CLI Command Reference.

## set-type-configuration

Il seguente esempio di codice mostra come utilizzare `set-type-configuration`.

AWS CLI

Per configurare i dati

L'`set-type-configuration` esempio seguente specifica i dati di configurazione per un' CloudFormation estensione registrata, nell'account e nella regione specificati.

```
aws cloudformation set-type-configuration \  
  --region us-west-2 \  
  --type RESOURCE \  
  --type-name Example::Test::Type \  
  --configuration-alias default \  
  --configuration "{\"CredentialKey\": \"testUserCredential\"}"
```

## Output:

```
{
  "ConfigurationArn": "arn:aws:cloudformation:us-west-2:123456789012:type-
configuration/resource/Example-Test-Type/default"
}
```

Per ulteriori informazioni, vedere [Uso del AWS CloudFormation registro](#) nella Guida per l'AWS CloudFormation utente.

- Per API i dettagli, vedere [SetTypeConfiguration](#) in AWS CLI Command Reference.

## set-type-default-version

Il seguente esempio di codice mostra come utilizzare `set-type-default-version`.

### AWS CLI

Per impostare la versione predefinita di un tipo

L'`set-type-default-version` esempio seguente imposta la versione del tipo specificato da utilizzare come predefinita per questo tipo.

```
aws cloudformation set-type-default-version \
  --type RESOURCE \
  --type-name My::Logs::LogGroup \
  --version-id 00000003
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Using the CloudFormation Registry](#) nella AWS CloudFormation Users Guide.

- Per API i dettagli, vedere [SetTypeDefaultVersion](#) in AWS CLI Command Reference.

## signal-resource

Il seguente esempio di codice mostra come utilizzare `signal-resource`.

### AWS CLI

Per segnalare una risorsa

L'`signal-resource`esempio seguente segnala success di soddisfare la condizione di attesa indicata `MyWaitCondition` nello stack denominato `my-stack`.

```
aws cloudformation signal-resource \  
  --stack-name my-stack \  
  --logical-resource-id MyWaitCondition \  
  --unique-id 1234 \  
  --status SUCCESS
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [SignalResource](#) in AWS CLI Command Reference.

## stop-stack-set-operation

Il seguente esempio di codice mostra come utilizzare `stop-stack-set-operation`.

### AWS CLI

Per interrompere un'operazione di stack set

L'`stop-stack-set-operation`esempio seguente interrompe un'operazione di aggiornamento in corso sul set di stack specificato.

```
aws cloudformation stop-stack-set-operation \  
  --stack-set-name my-stack-set \  
  --operation-id 1261cd27-490b-xmpl-ab42-793a896c69e6
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [StopStackSetOperation](#) in AWS CLI Command Reference.

## test-type

Il seguente esempio di codice mostra come utilizzare `test-type`.

### AWS CLI

Per testare un'estensione

L'`test-type`esempio seguente verifica un'estensione registrata per assicurarsi che soddisfi tutti i requisiti necessari per la pubblicazione nel CloudFormation registro.

```
aws cloudformation test-type \  
  --arn arn:aws:cloudformation:us-west-2:123456789012:type/resource/Sample-Test-Resource123/00000001
```

Output:

```
{  
  "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/  
Sample-Test-Resource123/00000001"  
}
```

Per ulteriori informazioni, vedere [Uso del AWS CloudFormation registro](#) nella Guida per l'AWS CloudFormation utente.

- Per API i dettagli, vedere [TestType](#) in AWS CLI Command Reference.

## update-stack-instances

Il seguente esempio di codice mostra come utilizzare `update-stack-instances`.

AWS CLI

Per aggiornare le istanze dello stack

L'update-stack-instancesesempio seguente riprova a eseguire un aggiornamento sulle istanze dello stack in due account in due regioni con le impostazioni più recenti. L'impostazione di tolleranza agli errori specificata garantisce che l'aggiornamento venga tentato in tutti gli account e le regioni, anche se alcuni stack non possono essere aggiornati.

```
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
```

Output:

```
{  
  "OperationId": "103ebdf2-21ea-xmpl-8892-de5e30733132"  
}
```

- Per API i dettagli, vedere [UpdateStackInstances](#) in AWS CLI Command Reference.

## update-stack-set

Il seguente esempio di codice mostra come utilizzare `update-stack-set`.

### AWS CLI

Per aggiornare un set di stack

L'update-stack-set esempio seguente aggiunge un tag con il nome della chiave `Owner` e il valore di `IT` alle istanze dello stack nel set di stack specificato.

```
aws cloudformation update-stack-set \  
  --stack-set-name my-stack-set \  
  --use-previous-template \  
  --tags Key=Owner,Value=IT
```

Output:

```
{  
  "OperationId": "e2b60321-6cab-xmpl-bde7-530c6f47950e"  
}
```

- Per API i dettagli, vedere [UpdateStackSet](#) in AWS CLI Command Reference.

## update-stack

Il seguente esempio di codice mostra come utilizzare `update-stack`.

### AWS CLI

Per aggiornare gli AWS CloudFormation stack

Il `update-stack` comando seguente aggiorna il modello e i parametri di input per lo `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
```

Il `update-stack` comando seguente aggiorna solo il valore del `SubnetIDs` parametro per lo `mystack` stack. Se non si specifica un valore di parametro, viene utilizzato il valore predefinito specificato nel modello:

```
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
```

Il `update-stack` comando seguente aggiunge due argomenti di notifica dello stack allo `mystack` stack:

```
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"
```

Per ulteriori informazioni, consulta [gli aggiornamenti AWS CloudFormation dello stack nella Guida per l'AWS CloudFormation utente](#).

- Per API i dettagli, vedere [UpdateStack](#) in AWS CLI Command Reference.

## update-termination-protection

Il seguente esempio di codice mostra come utilizzare `update-termination-protection`.

AWS CLI

Per abilitare la protezione dalla terminazione

L'`update-termination-protection` esempio seguente abilita la protezione dalla terminazione nello stack specificato.

```
aws cloudformation update-termination-protection \  
--stack-name my-stack \  
--enable-termination-protection
```

Output:

```
{
  "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204"
}
```

- Per API i dettagli, vedere [UpdateTerminationProtection](#) in AWS CLI Command Reference.

## validate-template

Il seguente esempio di codice mostra come utilizzare `validate-template`.

### AWS CLI

Per convalidare un modello AWS CloudFormation

Il `validate-template` comando seguente convalida il `sampletemplate.json` modello:

```
aws cloudformation validate-template --template-body file://sampletemplate.json
```

Output:

```
{
  "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": []
}
```

Per ulteriori informazioni, vedere [Working with AWS CloudFormation Templates](#) nella Guida per l'AWS CloudFormation utente.

- Per API i dettagli, vedere [ValidateTemplate](#) in AWS CLI Command Reference.

## CloudFront esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with CloudFront.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-cloud-front-origin-access-identity**

Il seguente esempio di codice mostra come utilizzare `create-cloud-front-origin-access-identity`.

AWS CLI

Per creare un'identità di accesso all' CloudFront origine

L'esempio seguente crea un'identità di accesso all' CloudFront origine (OAI) fornendo la OAI configurazione come argomento della riga di comando:

```
aws cloudfront create-cloud-front-origin-access-identity \  
  --cloud-front-origin-access-identity-config \  
    CallerReference="cli-example",Comment="Example OAI"
```

È possibile eseguire la stessa operazione fornendo la OAI configurazione in un JSON file, come illustrato nell'esempio seguente:

```
aws cloudfront create-cloud-front-origin-access-identity \  
  --cloud-front-origin-access-identity-config file://OAI-config.json
```

Il file `OAI-config.json` è un JSON documento nella directory corrente che contiene quanto segue:

```
{  
  "CallerReference": "cli-example",
```

```
"Comment": "Example OAI"
}
```

Sia che si fornisca alla OAI configurazione un argomento della riga di comando o un JSON file, l'output è lo stesso:

```
{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/origin-access-identity/
cloudfront/E74FTE3AEXAMPLE",
  "ETag": "E2QWRUHEXAMPLE",
  "CloudFrontOriginAccessIdentity": {
    "Id": "E74FTE3AEXAMPLE",
    "S3CanonicalUserId":
"cd13868f797c227f7bea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
    "CloudFrontOriginAccessIdentityConfig": {
      "CallerReference": "cli-example",
      "Comment": "Example OAI"
    }
  }
}
```

- Per API i dettagli, vedere [CreateCloudFrontOriginAccessIdentity](#) in AWS CLI Command Reference.

## create-distribution-with-tags

Il seguente esempio di codice mostra come utilizzare `create-distribution-with-tags`.

### AWS CLI

Per creare una CloudFront distribuzione con tag

L'esempio seguente crea una distribuzione con due tag fornendo la configurazione di distribuzione e i tag in un JSON file denominato `dist-config-with-tags.json`:

```
aws cloudfront create-distribution-with-tags \
  --distribution-config-with-tags file://dist-config-with-tags.json
```

Il file `dist-config-with-tags.json` è un JSON documento nella cartella corrente che contiene quanto segue. Notate l'Tags oggetto nella parte superiore del file, che contiene due tag:

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
  }
}

```

### Output:

```

{
  "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
  }
}

```

- Per API i dettagli, consultate [CreateDistributionWithTags AWS CLI Command Reference](#).

## create-distribution

Il seguente esempio di codice mostra come utilizzare `create-distribution`.

### AWS CLI

Per creare una CloudFront distribuzione

L'esempio seguente crea una distribuzione per un bucket S3 denominato `awsexamplebucket` e lo specifica anche `index.html` come oggetto root predefinito, utilizzando argomenti della riga di comando:

```

aws cloudfront create-distribution \
  --origin-domain-name awsexamplebucket.s3.amazonaws.com \
  --default-root-object index.html

```

Invece di utilizzare gli argomenti della riga di comando, è possibile fornire la configurazione della distribuzione in un JSON file, come illustrato nell'esempio seguente:

```

aws cloudfront create-distribution \
  --distribution-config file://dist-config.json

```

Il file `dist-config.json` è un JSON documento nella cartella corrente che contiene quanto segue:

```
{
  "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
}

```

Sia che si forniscano le informazioni sulla distribuzione con un argomento della riga di comando o un JSON file, l'output è lo stesso:

```

{
  "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
}
}
```

- Per API i dettagli, vedere [CreateDistribution](#) in AWS CLI Command Reference.

## create-field-level-encryption-config

Il seguente esempio di codice mostra come utilizzare `create-field-level-encryption-config`.

### AWS CLI

Per creare una configurazione di CloudFront crittografia a livello di campo

L'esempio seguente crea una configurazione di crittografia a livello di campo fornendo i parametri di configurazione in un file denominato. JSON `fle-config.json` Prima di poter creare una configurazione di crittografia a livello di campo, è necessario disporre di un profilo di crittografia a livello di campo. Per creare un profilo, vedere il comando `-profile`. `create-field-level-encryption`

Per ulteriori informazioni sulla crittografia a CloudFront livello di campo, consulta [Using Field-Level Encryption to Help Protect Sensitive Data nella Amazon Developer Guide](#). CloudFront

```
aws cloudfront create-field-level-encryption-config \  
  --field-level-encryption-config file://fle-config.json
```

Il file `fle-config.json` è un JSON documento nella cartella corrente che contiene quanto segue:

```
{  
  "CallerReference": "cli-example",  
  "Comment": "Example FLE configuration",  
  "QueryArgProfileConfig": {  
    "ForwardWhenQueryArgProfileIsUnknown": true,  
    "QueryArgProfiles": {  
      "Quantity": 0  
    }  
  },  
  "ContentTypeProfileConfig": {  
    "ForwardWhenContentTypeIsUnknown": true,  
    "ContentTypeProfiles": {  
      "Quantity": 1,  
      "Items": [  
        {  
          "Format": "URLEncoded",
```



```

        "ProfileId": "P280MFCLSY0CVU",
        "ContentType": "application/x-www-form-urlencoded"
      }
    ]
  }
}

```

### Output:

```

{
  "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": "P280MFCLSY0CVU",
              "ContentType": "application/x-www-form-urlencoded"
            }
          ]
        }
      }
    }
  }
}

```

- Per API i dettagli, vedere [CreateFieldLevelEncryptionConfigin AWS CLI Command Reference](#).

## create-field-level-encryption-profile

Il seguente esempio di codice mostra come utilizzare `create-field-level-encryption-profile`.

### AWS CLI

Per creare un profilo di CloudFront crittografia a livello di campo

L'esempio seguente crea un profilo di crittografia a livello di campo fornendo i parametri in un file denominato `fle-profile-config.json`. Prima di poter creare un profilo di crittografia a livello di campo, è necessario disporre di una chiave pubblica. CloudFront Per creare una chiave CloudFront pubblica, vedi il `create-public-key` comando.

Per ulteriori informazioni sulla crittografia a CloudFront livello di campo, consulta [Using Field-Level Encryption to Help Protect Sensitive Data nella Amazon Developer Guide](#). CloudFront

```
aws cloudfront create-field-level-encryption-profile \
  --field-level-encryption-profile-config file://fle-profile-config.json
```

Il file `fle-profile-config.json` è un JSON documento nella cartella corrente che contiene quanto segue:

```
{
  "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"
          ]
        }
      ]
    ]
  }
}
```

```

    }
  ]
}

```

Output:

```

{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/field-level-encryption-
profile/PPK0U0SIF5WSV",
  "ETag": "E2QWRUHEXAMPLE",
  "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"
              ]
            }
          }
        ]
      }
    }
  }
}

```

- Per API i dettagli, vedere [CreateFieldLevelEncryptionProfile](#) in AWS CLI Command Reference.

## create-invalidation

Il seguente esempio di codice mostra come utilizzare `create-invalidation`.

## AWS CLI

Per creare un'invalidazione per una distribuzione CloudFront

L'`create-invalidation` seguente crea un'invalidazione per i file specificati nella distribuzione specificata: CloudFront

```
aws cloudfront create-invalidation \  
  --distribution-id EDFDVBD6EXAMPLE \  
  --paths "/example-path/example-file.jpg" "/example-path/example-file2.png"
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Nell'esempio precedente, generavano AWS CLI automaticamente un messaggio casuale. `CallerReference` Per specificarne uno personalizzato `CallerReference` o per evitare di passare i parametri di invalidazione come argomenti della riga di comando, è possibile utilizzare un JSON file. L'esempio seguente crea un'invalidazione per due file, fornendo i parametri di invalidazione in un file denominato: `JSON inv-batch.json`

```
aws cloudfront create-invalidation \  
  --distribution-id EDFDVBD6EXAMPLE \  
  --paths "/example-path/example-file.jpg" "/example-path/example-file2.png"
```

```
--invalidation-batch file://inv-batch.json
```

Contenuto di `inv-batch.json`.

```
{
  "Paths": {
    "Quantity": 2,
    "Items": [
      "/example-path/example-file.jpg",
      "/example-path/example-file2.png"
    ]
  },
  "CallerReference": "cli-example"
}
```

Output:

```
{
  "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"
    }
  }
}
```

- Per i API dettagli, vedere [CreateInvalidation](#) in Command Reference.AWS CLI

## create-public-key

Il seguente esempio di codice mostra come utilizzare `create-public-key`.

## AWS CLI

Per creare una chiave CloudFront pubblica

L'esempio seguente crea una chiave CloudFront pubblica fornendo i parametri in un JSON file denominato `pub-key-config.json`. Prima di poter utilizzare questo comando, è necessario disporre di una chiave pubblica PEM con codifica. Per ulteriori informazioni, consulta [Create an RSA Key Pair](#) nella Amazon CloudFront Developer Guide.

```
aws cloudfront create-public-key \
  --public-key-config file://pub-key-config.json
```

Il file `pub-key-config.json` è un JSON documento nella cartella corrente che contiene quanto segue. Si noti che la chiave pubblica è codificata nel PEM formato.

```
{
  "CallerReference": "cli-example",
  "Name": "ExampleKey",
  "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAA0CAQ8AMIIBcGKCAQEAXPmBCA2Ks0lnd7IR+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"
}
```

Output:

```
{
  "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-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAA0CAQ8AMIIBCgKCAQEAxPMbCA2Ks01nd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesp1c0kjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6Zq1V3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nnq
+kGZ2NQ0FyIyT2eiLK0X5RgB/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nnrwIDAQAB\n-----END
PUBLIC KEY-----\n",
    "Comment": "example public key"
  }
}
}

```

- Per API i dettagli, vedere [CreatePublicKey](#) in AWS CLI Command Reference.

## delete-cloud-front-origin-access-identity

Il seguente esempio di codice mostra come utilizzare `delete-cloud-front-origin-access-identity`.

### AWS CLI

Per eliminare un'identità di accesso di CloudFront origine

L'esempio seguente elimina l'identità di accesso di origine (OAI) con l'ID. `E74FTE3AEXAMPLE`. Per eliminare un'OAI, devi avere l'OAIID eETag. L'OAIID viene restituito nell'output dei comandi `create-cloud-front-origin -access-identity` e `list-cloud-front-origin -access-identities`. Per ottenere il, usa il comando `-access-identity ETag` o `-. get-cloud-front-origin get-cloud-front-origin access-identity-config` Usa l'`--if-match` opzione per fornire i. OAI ETag

```

aws cloudfront delete-cloud-front-origin-access-identity \
  --id E74FTE3AEXAMPLE \
  --if-match E2QWRUHEXAMPLE

```

In caso di successo, questo comando non ha alcun risultato.

- Per API i dettagli, vedere [DeleteCloudFrontOriginAccessIdentity](#) in AWS CLI Command Reference.

## delete-distribution

Il seguente esempio di codice mostra come utilizzare `delete-distribution`.

### AWS CLI

Per eliminare una CloudFront distribuzione

L'esempio seguente elimina la CloudFront distribuzione con l'ID. EDFDVBD6EXAMPLE Prima di poter eliminare una distribuzione, è necessario disattivarla. Per disabilitare una distribuzione, usa il comando `update-distribution`. Per ulteriori informazioni, consulta gli esempi di `update-distribution`.

Quando una distribuzione è disabilitata, puoi eliminarla. Per eliminare una distribuzione, è necessario utilizzare l'opzione `--if-match` per fornire la distribuzione ETag. Per ottenere la ETag, usa il comando `get-distribution` o `get-distribution-config`.

```
aws cloudfront delete-distribution \  
  --id EDFDVBD6EXAMPLE \  
  --if-match E2QWRUHEXAMPLE
```

In caso di successo, questo comando non produce alcun risultato.

- Per API i dettagli, vedere [DeleteDistribution](#) in AWS CLI Command Reference.

## delete-field-level-encryption-config

Il seguente esempio di codice mostra come utilizzare `delete-field-level-encryption-config`.

### AWS CLI

Per eliminare una configurazione di CloudFront crittografia a livello di campo

L'esempio seguente elimina la configurazione di crittografia a CloudFront livello di campo con l'ID. C3KM2WVD605UAY Per eliminare una configurazione di crittografia a livello di campo, è necessario disporre del relativo ID e. ETag L'ID viene restituito nell'output dei comandi `create-field-level-encryption -config` e `-configs`. `list-field-level-encryption` Per ottenere il ETag, usa il comando `or -config`. `get-field-level-encryption` `get-field-level-encryption` Usa l'opzione `--if-match` per fornire le configurazioni. ETag

```
aws cloudfront delete-field-level-encryption-config \  
  --id C3KM2WVD605UAY
```



```
--id C3KM2WVD605UAY \  
--if-match E26M4BIAV81ZF6
```

In caso di successo, questo comando non ha alcun risultato.

- Per API i dettagli, vedere [DeleteFieldLevelEncryptionConfig](#) in AWS CLI Command Reference.

## delete-field-level-encryption-profile

Il seguente esempio di codice mostra come utilizzare `delete-field-level-encryption-profile`.

### AWS CLI

Per eliminare un profilo di CloudFront crittografia a livello di campo

L'esempio seguente elimina il profilo di crittografia a CloudFront livello di campo con l'ID. PPK0U0SIF5WSV Per eliminare un profilo di crittografia a livello di campo, è necessario disporre del relativo ID e. ETag L'ID viene restituito nell'output dei comandi `create-field-level-encryption-profile` e `list-field-level-encryption-profiles`. Per ottenere il ETag, usa il comando `get-field-level-encryption-profile` o `get-field-level-encryption-profile-config`. Usa l' `--if-match` opzione per fornire i profili. ETag

```
aws cloudfront delete-field-level-encryption-profile \  
--id PPK0U0SIF5WSV \  
--if-match EJETYFJ9CL66D
```

In caso di successo, questo comando non ha alcun risultato.

- Per API i dettagli, vedere [DeleteFieldLevelEncryptionProfile](#) in AWS CLI Command Reference.

## delete-public-key

Il seguente esempio di codice mostra come utilizzare `delete-public-key`.

### AWS CLI

Per eliminare una chiave CloudFront pubblica

L'esempio seguente elimina la chiave CloudFront pubblica con l'ID. KDFB19YGCR002 Per eliminare una chiave pubblica, è necessario disporre del relativo ID e ETag. L'ID viene restituito

nell'output dei `list-public-keys` comandi `create-public-key` and. Per ottenere il `ETag`, usa il `get-public-key-config` comando `get-public-key` or. Usa l' `--if-match` opzione per fornire le chiavi pubbliche `ETag`.

```
aws cloudfront delete-public-key \  
  --id KDFB19YGCR002 \  
  --if-match E2QWRUHEXAMPLE
```

In caso di successo, questo comando non ha alcun risultato.

- Per API i dettagli, vedere [DeletePublicKey](#) in AWS CLI Command Reference.

## get-cloud-front-origin-access-identity-config

Il seguente esempio di codice mostra come utilizzare `get-cloud-front-origin-access-identity-config`.

### AWS CLI

Per ottenere una configurazione dell'identità di accesso all' CloudFront origine

L'esempio seguente ottiene i metadati sull'identità di accesso di CloudFront origine (OAI) con l'`E74FTE3AEXAMPLE`, incluso il relativo `ETag`. L'`OAIID` viene restituito nell'output dei comandi `create-cloud-front-origin-access-identity` e `-access-identities`. `list-cloud-front-origin`

```
aws cloudfront get-cloud-front-origin-access-identity-config --id E74FTE3AEXAMPLE
```

Output:

```
{  
  "ETag": "E2QWRUHEXAMPLE",  
  "CloudFrontOriginAccessIdentityConfig": {  
    "CallerReference": "cli-example",  
    "Comment": "Example OAI"  
  }  
}
```

- Per API i dettagli, vedere in Command Reference.  
[GetCloudFrontOriginAccessIdentityConfig](#) AWS CLI

## get-cloud-front-origin-access-identity

Il seguente esempio di codice mostra come utilizzare `get-cloud-front-origin-access-identity`.

### AWS CLI

Per ottenere un'identità di accesso all' CloudFront origine

L'esempio seguente ottiene l'identità di accesso di CloudFront origine (OAI) con l'ID `E74FTE3AEXAMPLE`, incluso il relativo ID canonico S3 ETag e l'ID canonico S3 associato. L'OAIID viene restituito nell'output dei comandi `-access-identity` e `create-cloud-front-origin-access-identities`. `list-cloud-front-origin`

```
aws cloudfront get-cloud-front-origin-access-identity --id E74FTE3AEXAMPLE
```

Output:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "CloudFrontOriginAccessIdentity": {
    "Id": "E74FTE3AEXAMPLE",
    "S3CanonicalUserId":
    "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
    "CloudFrontOriginAccessIdentityConfig": {
      "CallerReference": "cli-example",
      "Comment": "Example OAI"
    }
  }
}
```

- Per API i dettagli, vedere in Command Reference. [GetCloudFrontOriginAccessIdentity](#) AWS CLI

## get-distribution-config

Il seguente esempio di codice mostra come utilizzare `get-distribution-config`.

### AWS CLI

Per ottenere una configurazione CloudFront di distribuzione

L'esempio seguente ottiene i metadati sulla CloudFront distribuzione con l'ID `EDFDVBD6EXAMPLE`, incluso il relativo ETag. L'ID di distribuzione viene restituito nei comandi `create-distribution` e `list-distribution`.

```
aws cloudfront get-distribution-config --id EDFDVBD6EXAMPLE
```

Output:

```
{
  "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
  },
  "CustomErrorResponse": {
    "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
}
}
```

- Per i API dettagli, vedere in Command Reference. [GetDistributionConfig](#) AWS CLI

## get-distribution

Il seguente esempio di codice mostra come utilizzare `get-distribution`.

### AWS CLI

Per ottenere una CloudFront distribuzione

L'esempio seguente ottiene la CloudFront distribuzione con l'ID `EDFDVBD6EXAMPLE`, incluso il relativo ETag. L'ID di distribuzione viene restituito nei comandi `create-distribution` e `list-distribution`.

```
aws cloudfront get-distribution --id EDFDVBD6EXAMPLE
```

Output:

```
{
  "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
  },
  "CustomErrorResponse": {
    "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
    }
}
}
}

```

- Per i API dettagli, vedere in Command Reference. [GetDistribution](#) AWS CLI

## get-field-level-encryption-config

Il seguente esempio di codice mostra come utilizzare `get-field-level-encryption-config`.

### AWS CLI

Per ottenere i metadati relativi a una configurazione di crittografia a CloudFront livello di campo

L'esempio seguente ottiene i metadati relativi alla configurazione di crittografia a CloudFront livello di campo con l'ID, inclusi: C3KM2WVD605UAY ETag

```
aws cloudfront get-field-level-encryption-config --id C3KM2WVD605UAY
```

Output:

```
{
```

```

    "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": "P280MFCLSYOCVU",
              "ContentType": "application/x-www-form-urlencoded"
            }
          ]
        }
      }
    }
  }
}

```

- Per API i dettagli, vedere [GetFieldLevelEncryptionConfig](#) in AWS CLI Command Reference.

## get-field-level-encryption-profile-config

Il seguente esempio di codice mostra come utilizzare `get-field-level-encryption-profile-config`.

### AWS CLI

Per ottenere una configurazione del profilo di crittografia CloudFront a livello di campo

L'esempio seguente ottiene i metadati relativi al profilo di crittografia a CloudFront livello di campo con ID, inclusi: PPK0U0SIF5WSV ETag

```
aws cloudfront get-field-level-encryption-profile-config --id PPK0U0SIF5WSV
```

**Output:**

```
{
  "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"
            ]
          }
        }
      ]
    }
  }
}
```

- Per API i dettagli, vedere [GetFieldLevelEncryptionProfileConfig](#) in AWS CLI Command Reference.

**get-field-level-encryption-profile**

Il seguente esempio di codice mostra come utilizzare `get-field-level-encryption-profile`.

**AWS CLI**

Per ottenere un profilo di CloudFront crittografia a livello di campo

L'esempio seguente ottiene il profilo di crittografia a CloudFront livello di campo con ID `PPK0U0SIF5WSV`, incluso il relativo: ETag

```
aws cloudfront get-field-level-encryption-profile --id PPK0U0SIF5WSV
```

**Output:**

```
{
  "ETag": "E1QQG65FS2L2GC",
  "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"
              ]
            }
          ]
        ]
      }
    }
  }
}
```

- Per API i dettagli, vedere [GetFieldLevelEncryptionProfile](#) in AWS CLI Command Reference.

**get-field-level-encryption**

Il seguente esempio di codice mostra come utilizzare `get-field-level-encryption`.

**AWS CLI**

Per ottenere una configurazione di CloudFront crittografia a livello di campo

L'esempio seguente ottiene la configurazione di crittografia a CloudFront livello di campo con l'ID, inclusa la relativa: C3KM2WVD605UAY ETag

```
aws cloudfront get-field-level-encryption --id C3KM2WVD605UAY
```

Output:

```
{
  "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"
            }
          ]
        }
      }
    }
  }
}
```

- Per API i dettagli, vedere [GetFieldLevelEncryption](#) in AWS CLI Command Reference.

## get-invalidations

Il seguente esempio di codice mostra come utilizzare `get-invalidations`.

## AWS CLI

Per ottenere un' CloudFront invalidazione

L'esempio seguente ottiene l'invalidazione con l'ID I2J0I21PCUY0IK per la CloudFront distribuzione con l'ID: EDFDVBD6EXAMPLE

```
aws cloudfront get-invalidation --id I2J0I21PCUY0IK --distribution-id EDFDVBD6EXAMPLE
```

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [GetInvalidation](#) in AWS CLI Command Reference.

### get-public-key-config

Il seguente esempio di codice mostra come utilizzare `get-public-key-config`.

## AWS CLI

Per ottenere una configurazione a chiave CloudFront pubblica

L'esempio seguente ottiene i metadati sulla chiave CloudFront pubblica con l'IDKDFB19YGCR002, incluso il relativo ETag. L'ID della chiave pubblica viene restituito nei list-public-keys comandi create-public-key and.

```
aws cloudfront get-public-key-config --id KDFB19YGCR002
```

Output:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "PublicKeyConfig": {
    "CallerReference": "cli-example",
    "Name": "ExampleKey",
    "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPmCA2Ks01nd7IR+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"
  }
}
```

- Per API i dettagli, vedere [GetPublicKeyConfig](#) in AWS CLI Command Reference.

## get-public-key

Il seguente esempio di codice mostra come utilizzare get-public-key.

### AWS CLI

Per ottenere una chiave CloudFront pubblica

L'esempio seguente ottiene la chiave CloudFront pubblica con l'IDKDFB19YGCR002, incluso il relativo ETag. L'ID della chiave pubblica viene restituito nei list-public-keys comandi create-public-key and.

```
aws cloudfront get-public-key --id KDFB19YGCR002
```

**Output:**

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "PublicKey": {
    "Id": "KDFB19YGCR002",
    "CreatedTime": "2019-12-05T18:51:43.781Z",
    "PublicKeyConfig": {
      "CallerReference": "cli-example",
      "Name": "ExampleKey",
      "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAA0CAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+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"
    }
  }
}
```

- Per API i dettagli, vedere [GetPublicKey](#) in AWS CLI Command Reference.

**list-cloud-front-origin-access-identities**

Il seguente esempio di codice mostra come utilizzare `list-cloud-front-origin-access-identities`.

**AWS CLI**

Per elencare le identità di accesso all' CloudFront origine

L'esempio seguente ottiene un elenco delle identità di accesso di CloudFront origine (OAI) presenti nel tuo AWS account:

```
aws cloudfront list-cloud-front-origin-access-identities
```

**Output:**

```
{
```



```
"CloudFrontOriginAccessIdentityList": {
  "Items": [
    {
      "Id": "E74FTE3AEXAMPLE",
      "S3CanonicalUserId":
"cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
      "Comment": "Example OAI"
    },
    {
      "Id": "EH1HDMBEXAMPLE",
      "S3CanonicalUserId":
"1489f6f2e6faacaae7ff64c4c3e6956c24f78788abfc1718c3527c263bf7a17EXAMPLE",
      "Comment": "Test OAI"
    },
    {
      "Id": "E2X2C9TEXAMPLE",
      "S3CanonicalUserId":
"cbfeebb915a64749f9be546a45b3fcfd3a31c779673c13c4dd460911ae402c2EXAMPLE",
      "Comment": "Example OAI #2"
    }
  ]
}
```

- Per API i dettagli, vedere [ListCloudFrontOriginAccessIdentities](#) in AWS CLI Command Reference.

## list-distributions

Il seguente esempio di codice mostra come utilizzare `list-distributions`.

### AWS CLI

Per elencare le CloudFront distribuzioni

L'esempio seguente ottiene un elenco delle CloudFront distribuzioni presenti nel tuo AWS account:

```
aws cloudfront list-distributions
```

Output:

```

{
  "DistributionList": {
    "Items": [
      {
        "Id": "EMLARXS9EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/
EMLARXS9EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-11-22T00:55:15.705Z",
        "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": {

```

```
example",
    "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
},
"CustomErrorResponses": {
```



```

        "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
    }
]
}
}
}

```

- Per API i dettagli, vedere [ListDistributions](#) in AWS CLI Command Reference.

## list-field-level-encryption-configs

Il seguente esempio di codice mostra come utilizzare `list-field-level-encryption-configs`.

### AWS CLI

Per elencare le configurazioni di CloudFront crittografia a livello di campo

L'esempio seguente ottiene un elenco delle configurazioni di crittografia a CloudFront livello di campo presenti nell'account: AWS

```
aws cloudfront list-field-level-encryption-configs
```

Output:

```
{
```

```

"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": "P280MFCLSYOCVU",
              "ContentType": "application/x-www-form-urlencoded"
            }
          ]
        }
      }
    }
  ]
}

```

- Per API i dettagli, vedere [ListFieldLevelEncryptionConfigs](#) in AWS CLI Command Reference.

## list-field-level-encryption-profiles

Il seguente esempio di codice mostra come utilizzare `list-field-level-encryption-profiles`.

### AWS CLI

Per elencare i profili di CloudFront crittografia a livello di campo

L'esempio seguente ottiene un elenco dei profili di crittografia a CloudFront livello di campo presenti nell'account: AWS

```
aws cloudfront list-field-level-encryption-profiles
```

Output:

```
{
  "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": "PPK0U0SIF5WSV",
        "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"
}
]
}
}

```

- Per API i dettagli, vedere [ListFieldLevelEncryptionProfiles](#) in AWS CLI Command Reference.

## list-invalidations

Il seguente esempio di codice mostra come utilizzare `list-invalidations`.

AWS CLI

Per elencare le CloudFront invalidazioni

L'esempio seguente ottiene un elenco delle invalidazioni per la CloudFront distribuzione con l'ID: EDFDVBDGEXAMPLE

```
aws cloudfront list-invalidations --distribution-id EDFDVBDGEXAMPLE
```

Output:

```

{
  "InvalidationList": {
    "Marker": "",
    "Items": [
      {
        "Status": "Completed",
        "Id": "YNY2LI2BVJ4NJU",
        "CreateTime": "2019-08-31T21:15:52.042Z"
      }
    ],
  },
}

```

```

    "IsTruncated": false,
    "MaxItems": 100,
    "Quantity": 1
  }
}

```

- Per API i dettagli, vedere [ListInvalidations](#) in AWS CLI Command Reference.

## list-public-keys

Il seguente esempio di codice mostra come utilizzare `list-public-keys`.

### AWS CLI

Per elencare le chiavi CloudFront pubbliche

L'esempio seguente ottiene un elenco delle chiavi CloudFront pubbliche del tuo AWS account:

```
aws cloudfront list-public-keys
```

Output:

```

{
  "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-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAp0CAg88A8+f4dujn9Izt
\n26LxtgAkn2opGgo/NKpMiaisyw5qlg3f1gol7FV6pYN178iJg3E08JBbwt1H
+cR9\nLGSf60NDeVhm760c39Np/vWg0dsGQcRbi9WmKZeS0DqjQGzVZWqPmito3FzWV6b
\nfVY5N36U/RdbVAJm95Km+qaMY1bIdF40t72bi3IkKYV5h1B2XoDjlQ9F6ajQKyTB
\nMHa3SN8q+3ZjQ4sJJ7D1V6r4wR8jDcFVD5NckWJmmgIVnk0QM37NYeoDnka0uTpu\nha/
+3b8t0b2z3LBVHPkp85zJRA0XacSwf5rZtPYKBNFsixTa2n55k2r218m0kMC4\nUwIDAQAB\n-----END
PUBLIC KEY-----",
        "Comment": "example public key #2"
    }
  ]
}

```

- Per API i dettagli, vedere [ListPublicKeys](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una CloudFront distribuzione

L'esempio seguente ottiene un elenco dei tag per una CloudFront distribuzione:

```

aws cloudfront list-tags-for-resource \
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE

```

Output:

```

{
  "Tags": {
    "Items": [
      {
        "Key": "DateCreated",
        "Value": "2019-12-04"
      },
      {
        "Key": "Name",

```

```

        "Value": "Example name"
      },
      {
        "Key": "Project",
        "Value": "Example project"
      }
    ]
  }
}

```

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## sign

Il seguente esempio di codice mostra come utilizzare `sign`.

### AWS CLI

Per firmare un CloudFront URL

L'esempio seguente firma un CloudFront URL. Per firmare un URL, è necessario l'ID della coppia di chiavi (chiamato Access Key ID nella AWS Management Console) e la chiave privata della coppia di chiavi del firmatario attendibile. CloudFront Per ulteriori informazioni su signedURLs, consulta [Serving Private Content with Signed URLs and Signed Cookies](#) nella 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

```

Output:

```

https://d111111abcdef8.cloudfront.net/private-content/private-
file.html?Expires=1577836800&Signature=nEXK7Kby47XKeZQKvc6pwkif6oZc-
JWSpDkH0UH7EBGGqvgurkeCbgL5VfUAXyLQuJxFwRQWscz-
owcq9KpmewCXrXQbPaJZNi9XSNwf4YKurPDQYaRQawKoenH0GFteRf9ELK-
Bs3nljTLjtbgzIUt7QJNKXcWr8AuUYikzGdJ4-qzx6WnxXfH~fxg4-
GGl6l2kgCpXUB6Jx6K~Y3kpV0dzUP0IqFLHAnJojbhxqrVejomZZ2XrquDvNUCCIbePGnR3d24UPaLXG4FK0qNEaWDIB
GNvjRJxqWf93uMobeM0iVYahb-e0KIItiQewGcm0eLZQ__&Key-Pair-Id=APKAEIBAERJR2EXAMPLE

```

- Per API i dettagli, consulta [Sign](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare una CloudFront distribuzione

L'`tag-resource` esempio seguente aggiunge due tag alla CloudFront distribuzione specificata.

```
aws cloudfront tag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tags 'Items=[{Key=Name,Value="Example name"},{Key=Project,Value="Example project"}]'
```

Invece di utilizzare argomenti della riga di comando, potete fornire i tag in un JSON file, come illustrato nell'esempio seguente:

```
aws cloudfront tag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tags file://tags.json
```

Contenuto di `tags.json`.

```
{  
  "Items": [  
    {  
      "Key": "Name",  
      "Value": "Example name"  
    },  
    {  
      "Key": "Project",  
      "Value": "Example project"  
    }  
  ]  
}
```

Questo comando non produce alcun output.



- Per API i dettagli, consultate [TagResource AWS CLI Command Reference](#).

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una CloudFront distribuzione

L'esempio seguente rimuove due tag da una CloudFront distribuzione utilizzando argomenti della riga di comando:

```
aws cloudfront untag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tag-keys Items=Name,Project
```

Invece di utilizzare gli argomenti della riga di comando, potete fornire le chiavi dei tag in un JSON file, come illustrato nell'esempio seguente:

```
aws cloudfront untag-resource \  
  --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \  
  --tag-keys file://tag-keys.json
```

Il file `tag-keys.json` è un JSON documento nella cartella corrente che contiene quanto segue:

```
{  
  "Items": [  
    "Name",  
    "Project"  
  ]  
}
```

In caso di successo, questo comando non produce alcun risultato.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-cloud-front-origin-access-identity

Il seguente esempio di codice mostra come utilizzare `update-cloud-front-origin-access-identity`.

## AWS CLI

Per aggiornare un'identità di accesso all' CloudFront origine

L'esempio seguente aggiorna l'identità di accesso di origine (OAI) con l'ID E74FTE3AEXAMPLE. L'unico campo che puoi aggiornare è OAI'sComment.

Per aggiornare un'OAI, devi avere OAI l'ID eETag. L'OAIID viene restituito nell'output dei comandi `create-cloud-front-origin -access-identity` e `list-cloud-front-origin -access-identities`. Per ottenere il, usa il comando `-access-identity ETag` o `-. get-cloud-front-origin get-cloud-front-origin access-identity-config` Usa l'`--if-match` opzione per fornire i. 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"
```

È possibile eseguire la stessa operazione fornendo la OAI configurazione in un JSON file, come mostrato nell'esempio seguente:

```
aws cloudfront update-cloud-front-origin-access-identity \  
  --id E74FTE3AEXAMPLE \  
  --if-match E2QWRUHEXAMPLE \  
  --cloud-front-origin-access-identity-config file://OAI-config.json
```

Il file `OAI-config.json` è un JSON documento nella directory corrente che contiene quanto segue:

```
{  
  "CallerReference": "cli-example",  
  "Comment": "Example OAI Updated"  
}
```

Sia che si fornisca alla OAI configurazione un argomento della riga di comando o un JSON file, l'output è lo stesso:

```
{  
  "ETag": "E9LHASXEXAMPLE",  
  "CloudFrontOriginAccessIdentity": {  
    "Id": "E74FTE3AEXAMPLE",
```

```

    "S3CanonicalUserId":
      "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
      "CloudFrontOriginAccessIdentityConfig": {
        "CallerReference": "cli-example",
        "Comment": "Example OAI Updated"
      }
    }
  }
}

```

- Per API i dettagli, vedere [UpdateCloudFrontOriginAccessIdentity](#) in AWS CLI Command Reference.

## update-distribution

Il seguente esempio di codice mostra come utilizzare `update-distribution`.

### AWS CLI

Per aggiornare l'oggetto radice predefinito di una CloudFront distribuzione

L'esempio seguente aggiorna l'oggetto root predefinito `index.html` per la CloudFront distribuzione con l'ID `EDFDVBD6EXAMPLE`:

```

aws cloudfront update-distribution --id EDFDVBD6EXAMPLE \
  --default-root-object index.html

```

Output:

```

{
  "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
},
"CustomErrorResponse": {
    "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
  }
}
}
}

```

Per aggiornare una CloudFront distribuzione

L'esempio seguente disabilita la CloudFront distribuzione con l'ID EMLARXS9EXAMPLE fornendo la configurazione della distribuzione in un JSON file denominato `dist-config-disable.json`. Per aggiornare una distribuzione, è necessario utilizzare l'opzione `--if-match` per fornire quella della ETag distribuzione. Per ottenere la ETag, usa il comando `get-distribution` o `get-distribution-config`.

Dopo aver utilizzato l'esempio seguente per disabilitare una distribuzione, è possibile utilizzare il comando `delete-distribution` per eliminarla.

```

aws cloudfront update-distribution \
  --id EMLARXS9EXAMPLE \
  --if-match E2QWRUHEXAMPLE \
  --distribution-config file://dist-config-disable.json

```

Il file `dist-config-disable.json` è un JSON documento nella cartella corrente che contiene quanto segue. Nota che il `Enabled` campo è impostato su `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
}

```

### Output:

```

{
  "ETag": "E9LHASXEXAMPLE",
  "Distribution": {
    "Id": "EMLARXS9EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
    "Status": "InProgress",
    "LastModifiedTime": "2019-12-06T18:32:35.553Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d111111abcdef8.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
}
}
```

```
}

```

- Per API i dettagli, vedere [UpdateDistribution](#) in AWS CLI Command Reference.

## update-field-level-encryption-config

Il seguente esempio di codice mostra come utilizzare `update-field-level-encryption-config`.

### AWS CLI

Per aggiornare una configurazione di CloudFront crittografia a livello di campo

L'esempio seguente aggiorna il `Comment` campo della configurazione di crittografia a livello di campo con l'ID `C3KM2WVD605UAY` fornendo i parametri in un file. JSON

Per aggiornare una configurazione di crittografia a livello di campo, è necessario disporre dell'ID della configurazione e. ETag L'ID viene restituito nell'output dei comandi `create-field-level-encryption-config` e `-configs.list-field-level-encryption` Per ottenere il ETag, usa il comando `or-config.get-field-level-encryption` `get-field-level-encryption` Usa l' `--if-match` opzione per fornire le configurazioni. ETag

```
aws cloudfront update-field-level-encryption-config \
  --id C3KM2WVD605UAY \
  --if-match E2P4Z4VU7TY5SG \
  --field-level-encryption-config file://fle-config.json
```

Il file `fle-config.json` è un JSON documento nella directory corrente che contiene quanto segue:

```
{
  "CallerReference": "cli-example",
  "Comment": "Updated example FLE configuration",
  "QueryArgProfileConfig": {
    "ForwardWhenQueryArgProfileIsUnknown": true,
    "QueryArgProfiles": {
      "Quantity": 0
    }
  },
  "ContentTypeProfileConfig": {
    "ForwardWhenContentTypeIsUnknown": true,
```

```

    "ContentTypeProfiles": {
      "Quantity": 1,
      "Items": [
        {
          "Format": "URLEncoded",
          "ProfileId": "P280MFCLSY0CVU",
          "ContentType": "application/x-www-form-urlencoded"
        }
      ]
    }
  }
}

```

### Output:

```

{
  "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": "P280MFCLSY0CVU",
            "ContentType": "application/x-www-form-urlencoded"
          }
        ]
      }
    }
  }
}

```

```

    }
  }
}

```

- Per API i dettagli, vedere [UpdateFieldLevelEncryptionConfig](#) in AWS CLI Command Reference.

## update-field-level-encryption-profile

Il seguente esempio di codice mostra come utilizzare `update-field-level-encryption-profile`.

### AWS CLI

Per aggiornare un profilo di CloudFront crittografia a livello di campo

L'esempio seguente aggiorna il profilo di crittografia a livello di campo con l'ID. `PPK0UOSIF5WSV`. Questo esempio aggiorna l'`Name` e `Comment` del profilo e aggiunge un secondo `FieldPatterns` elemento fornendo i parametri in un JSON file.

Per aggiornare un profilo di crittografia a livello di campo, è necessario disporre dell'ID del profilo e. ETag L'ID viene restituito nell'output dei comandi `create-field-level-encryption-profile` e `list-field-level-encryption-profiles`. Per ottenere il ETag, usa il comando `get-field-level-encryption-profile` o `get-field-level-encryption-profile-config`. Usa l'`--if-match` opzione per fornire i profili. ETag

```

aws cloudfront update-field-level-encryption-profile \
  --id PPK0UOSIF5WSV \
  --if-match E1QQG65FS2L2GC \
  --field-level-encryption-profile-config file://fle-profile-config.json

```

Il file `fle-profile-config.json` è un JSON documento nella directory corrente che contiene quanto segue:

```

{
  "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"
            ]
        }
    ]
}

```

### Output:

```

{
  "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"
              ]
            }
          }
        ]
      }
    }
  }
}

```

- Per API i dettagli, vedere [UpdateFieldLevelEncryptionProfile](#) in AWS CLI Command Reference.

## CloudSearch Esempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con Amazon CloudSearch.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **upload-documents**

Il seguente esempio di codice mostra come utilizzare `upload-documents`.

AWS CLI

Il `upload-documents` comando seguente carica un batch di JSON documenti in un CloudSearch dominio Amazon:

```
aws cloudsearchdomain upload-documents --endpoint-url https://doc-my-domain.us-west-1.cloudsearch.amazonaws.com --content-type application/json --documents document-batch.json
```

Output:

```
{
  "status": "success",
  "adds": 5000,
  "deletes": 0
}
```



- Per API i dettagli, consulta [UploadDocuments AWS CLI Command Reference](#).

## CloudTrail esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with CloudTrail.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **add-tags**

Il seguente esempio di codice mostra come utilizzare `add-tags`.

#### AWS CLI

Per aggiungere tag al percorso

Il `add-tags` comando seguente aggiunge tag per `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
```

- Per API i dettagli, vedere [AddTags](#) in AWS CLI Command Reference.

#### **create-subscription**

Il seguente esempio di codice mostra come utilizzare `create-subscription`.

## AWS CLI

Per creare e configurare AWS le risorse per un percorso

Il `create-subscription` comando seguente crea un nuovo bucket S3 e un nuovo SNS argomento per: `Trail1`

```
aws cloudtrail create-subscription --name Trail1 --s3-new-bucket my-bucket --sns-new-topic my-topic
```

Output:

```
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
```

- Per API i dettagli, consulta AWS CLI Command [CreateSubscriptionReference](#).

## **create-trail**

Il seguente esempio di codice mostra come utilizzare `create-trail`.

## AWS CLI

Per creare un percorso

Il `create-trail` comando seguente crea un percorso multiregionale denominato `Trail1` e specifica un bucket S3:

```
aws cloudtrail create-trail --name Trail1 --s3-bucket-name my-bucket --is-multi-region-trail
```

Output:

```
{
  "IncludeGlobalServiceEvents": true,
  "Name": "Trail1",
  "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
  "LogFileValidationEnabled": false,
  "IsMultiRegionTrail": true,
  "S3BucketName": "my-bucket"
}
```

- Per i API dettagli, consulta [CreateTrail](#) Command Reference.AWS CLI

## `delete-trail`

Il seguente esempio di codice mostra come utilizzare `delete-trail`.

## AWS CLI

Per eliminare una traccia

Il `delete-trail` comando seguente elimina una traccia denominata `Trail1`:

```
aws cloudtrail delete-trail --name Trail1
```

- Per API i dettagli, vedere [DeleteTrail](#) in AWS CLI Command Reference.

## `describe-trails`

Il seguente esempio di codice mostra come utilizzare `describe-trails`.

## AWS CLI

Per descrivere un percorso

Il `describe-trails` comando seguente restituisce le impostazioni per `Trail1` e `Trail2`:

```
aws cloudtrail describe-trails --trail-name-list Trail1 Trail2
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeTrails](#) in AWS CLI Command Reference.

## get-event-selectors

Il seguente esempio di codice mostra come utilizzare `get-event-selectors`.

### AWS CLI

Per visualizzare le impostazioni del selettore di eventi per un percorso

Il `get-event-selectors` comando seguente restituisce le impostazioni per `Trail1`:

```
aws cloudtrail get-event-selectors --trail-name Trail1
```

Output:

```
{
  "EventSelectors": [
    {
      "IncludeManagementEvents": true,
      "DataResources": [],
      "ReadWriteType": "All"
    }
  ],
  "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1"
}
```

- Per API i dettagli, vedere [GetEventSelectors](#) in AWS CLI Command Reference.

## get-trail-status

Il seguente esempio di codice mostra come utilizzare `get-trail-status`.

### AWS CLI

Per ottenere lo stato di un percorso

Il `get-trail-status` comando seguente restituisce i dettagli di consegna e registrazione per `Trail1`:

```
aws cloudtrail get-trail-status --name Trail1
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [GetTrailStatus](#) in AWS CLI Command Reference.

## list-public-keys

Il seguente esempio di codice mostra come utilizzare `list-public-keys`.

### AWS CLI

Per elencare tutte le chiavi pubbliche di un percorso

Il `list-public-keys` comando seguente restituisce tutte le chiavi pubbliche le cui chiavi private sono state utilizzate per firmare i file digest entro l'intervallo di tempo specificato:

```
aws cloudtrail list-public-keys --start-time 2016-01-01T20:30:00.000Z
```

Output:

```
{
  "PublicKeyList": [
    {
      "ValidityStartTime": 1453076702.0,
      "ValidityEndTime": 1455668702.0,
      "Value": "MIIBCgKCAQEAlSS3cl92HDycr/MTj0mo0has8habjrxaXw+Kz1WF0axSI2tcF
+3iJ9BKQAVSKxGwxwu3m0wG3J
+kU11xboEcEPHYoIYMbgfSw7KGNUdKwkLzsQWhUJ0cIb0HASox1vv/5fNXkrHhGbDCHeVXm804c83nvHUEFYThr1PfyP
+4WGDk+BGH5m9iuiAKkipEHWmU18/P7XpfpWQuk4h8g3pXZ0rNXr081bh4d39svj7Uqdhv0XoBISp9t/
EXYuePGEtBdrKD9Dz+VHwyUPtBQvYr9BnkF88qBnaPNhS44rzwIDAQAB",

```

```

    "Fingerprint": "7f3f401420072e50a65a141430817ab3"
  }
]
}

```

- Per API i dettagli, vedere [ListPublicKeys](#) in AWS CLI Command Reference.

## list-tags

Il seguente esempio di codice mostra come utilizzare `list-tags`.

### AWS CLI

Per elencare i tag di un percorso

Il `list-tags` comando seguente elenca i tag per `Trail1` e `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

```

Output:

```

{
  "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"
    }
  ]
}

```

- Per API i dettagli, vedere [ListTags](#) in AWS CLI Command Reference.

## lookup-events

Il seguente esempio di codice mostra come utilizzare `lookup-events`.

### AWS CLI

Per cercare gli eventi di un percorso

Il `lookup-events` comando seguente cerca gli eventi di API attività in base all'attributo `EventName`:

```
aws cloudtrail lookup-events --lookup-attributes AttributeKey=EventName,AttributeValue=ConsoleLogin
```

Output:

```

{
  "Events": [
    {
      "EventId": "654ccbc0-ba0d-486a-9076-dbf7274677a7",
      "Username": "my-session-name",
      "EventTime": "2021-11-18T09:41:02-08:00",
      "CloudTrailEvent": "{\n\"eventVersion\": \"1.02\", \n\"userIdentity\": {\n\"type\": \n\n\"AssumedRole\", \n\"principalId\": \"AR0AJIKPFTA72SWU4L7T4:my-session-name\", \n\"arn\": \n\n\"arn:aws:sts::123456789012:assumed-role/my-role/my-session-name\", \n\"accountId\": \n\n\"123456789012\", \n\"sessionContext\": {\n\"attributes\": {\n\"mfaAuthenticated\": \"false \n\n\", \n\"creationDate\": \"2016-01-26T21:42:12Z\"}, \n\"sessionIssuer\": {\n\"type\": \"Role\", \n\n\"principalId\": \"AR0AJIKPFTA72SWU4L7T4\", \n\"arn\": \"arn:aws:iam::123456789012:role/my-role\", \n\"accountId\": \"123456789012\", \n\"userName\": \"my-role\"}}}, \n\"eventTime \n\n\": \"2016-01-26T21:42:12Z\", \n\"eventSource\": \"signin.amazonaws.com\", \n\"eventName\": \n\n\"ConsoleLogin\", \n\"awsRegion\": \"us-east-1\", \n\"sourceIPAddress\": \"72.21.198.70\", \n\n\"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\": []
}
]
}
```

- Per API i dettagli, vedere [LookupEvents](#) in AWS CLI Command Reference.

## put-event-selectors

Il seguente esempio di codice mostra come utilizzare `put-event-selectors`.

### AWS CLI

Esempio 1: configura un percorso per registrare gli eventi di gestione e gli eventi relativi ai dati utilizzando selettori di eventi avanzati

È possibile aggiungere selettori di eventi avanzati e condizioni per i selettori di eventi avanzati, fino a un massimo di 500 valori per tutte le condizioni e i selettori su un percorso. È possibile utilizzare selettori di eventi avanzati per registrare tutti i tipi di eventi di dati disponibili. È possibile utilizzare selettori di eventi avanzati o selettori di eventi di base, ma non entrambi. Se si applicano selettori di eventi avanzati a un percorso, tutti i selettori di eventi di base esistenti vengono sovrascritti.

L'esempio seguente crea un selettore di eventi avanzato per un percorso denominato `myTrail` per registrare tutti gli eventi di gestione, registrare S3 `PutObject` e le `DeleteObject` API chiamate per tutti i bucket S3 tranne uno, registrare le chiamate dati API per una funzione Lambda denominata `e` e registrare le chiamate `Publish` API su un argomento denominato `myFunction`. SNS `myTopic`

```
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"] }]]}'
```

Output:

```
{
  "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"
        ]
    }
]
}
]
}

```

Per ulteriori informazioni, consulta [Registrazione degli eventi utilizzando selettori di eventi avanzati](#) nella Guida per l'utente.AWS CloudTrail

Esempio 2: configura i selettori di eventi per un percorso per registrare tutti gli eventi di gestione e gli eventi relativi ai dati

Per un trail puoi configurare fino a 5 selettori di eventi e un massimo di 250 risorse di dati. I selettori di eventi vengono anche chiamati selettori di eventi di base. Puoi utilizzare i selettori di eventi per registrare gli eventi di gestione e gli eventi relativi ai dati per oggetti S3, funzioni Lambda e tabelle DynamoDB. Per registrare gli eventi relativi ai dati per altri tipi di risorse, è necessario utilizzare selettori di eventi avanzati.

L'esempio seguente crea un selettore di eventi per un percorso denominato `TrailName` per includere tutti gli eventi di gestione, gli eventi di dati per due combinazioni di bucket/prefisso

Amazon S3 e gli eventi di dati per una singola funzione Lambda denominata. AWS 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"]}]]'
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Registrazione degli eventi utilizzando selettori di eventi di base nella Guida per l'utente AWS CloudTrail](#)

Esempio 3: configura i selettori di eventi per un trail per registrare gli eventi di gestione, tutti gli eventi dei dati S3 sugli oggetti S3 e tutti gli eventi dei dati Lambda sulle funzioni del tuo account

L'esempio seguente crea un selettore di eventi per un percorso denominato `TrailName2` che include tutti gli eventi di gestione e tutti gli eventi di dati per tutti i bucket Amazon S3 AWS e le funzioni Lambda nell'account. 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"]}]]'
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Registrazione degli eventi utilizzando selettori di eventi di base](#) nella Guida per l'utente AWS CloudTrail

- Per API i dettagli, vedere [PutEventSelectors](#) in AWS CLI Command Reference.

## remove-tags

Il seguente esempio di codice mostra come utilizzare `remove-tags`.

### AWS CLI

Per rimuovere i tag da un percorso

Il `remove-tags` comando seguente rimuove i tag specificati per `Trail1`:

```
aws cloudtrail remove-tags --resource-id arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 --tags-list Key=name Key=location
```

- Per API i dettagli, vedere [RemoveTags](#) in AWS CLI Command Reference.

## start-logging

Il seguente esempio di codice mostra come utilizzare `start-logging`.

### AWS CLI

Per iniziare a registrare un percorso

Il `start-logging` comando seguente attiva la registrazione per: `Trail1`

```
aws cloudtrail start-logging --name Trail1
```

- Per API i dettagli, vedere [StartLogging](#) in AWS CLI Command Reference.

## stop-logging

Il seguente esempio di codice mostra come utilizzare `stop-logging`.

### AWS CLI

Per interrompere la registrazione di un percorso

Il `stop-logging` comando seguente disattiva la registrazione per: `Trail1`

```
aws cloudtrail stop-logging --name Trail1
```

- Per API i dettagli, vedere [StopLogging](#) in AWS CLI Command Reference.

## update-subscription

Il seguente esempio di codice mostra come utilizzare `update-subscription`.

### AWS CLI

Per aggiornare le impostazioni di configurazione per un trail

Il `update-subscription` comando seguente aggiorna il percorso per specificare un nuovo bucket S3 e SNS un nuovo argomento:

```
aws cloudtrail update-subscription --name Trail1 --s3-new-bucket my-bucket-new --  
sns-new-topic my-topic-new
```

Output:

```
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"  
  }  
}
```

- Per API i dettagli, consulta AWS CLI Command [UpdateSubscription](#) Reference.



## update-trail

Il seguente esempio di codice mostra come utilizzare `update-trail`.

### AWS CLI

Per aggiornare una traccia

Il `update-trail` comando seguente aggiorna un trail per utilizzare un bucket esistente per la consegna dei log:

```
aws cloudtrail update-trail --name Trail1 --s3-bucket-name my-bucket
```

Output:

```
{
  "IncludeGlobalServiceEvents": true,
  "Name": "Trail1",
  "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
  "LogFileValidationEnabled": false,
  "IsMultiRegionTrail": true,
  "S3BucketName": "my-bucket"
}
```

- Per API i dettagli, vedere [UpdateTrail](#) in AWS CLI Command Reference.

## validate-logs

Il seguente esempio di codice mostra come utilizzare `validate-logs`.

### AWS CLI

Per convalidare un file di registro

Il `validate-logs` comando seguente convalida i log per: `Trail1`

```
aws cloudtrail validate-logs --trail-arn arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1 --start-time 20160129T19:00:00Z
```

Output:

```
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
```

- Per API i dettagli, vedere [ValidateLogs](#) in AWS CLI Command Reference.

## CloudWatch esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with CloudWatch.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

### Azioni

#### **delete-alarms**

Il seguente esempio di codice mostra come utilizzare `delete-alarms`.

AWS CLI

Per eliminare un allarme

L'esempio seguente utilizza il `delete-alarms` comando per eliminare l' CloudWatch allarme Amazon denominato «myalarm»:

```
aws cloudwatch delete-alarms --alarm-names myalarm
```

Output:

This command returns to the prompt if successful.

- Per API i dettagli, consulta [DeleteAlarms AWS CLI Command Reference](#).

## describe-alarm-history

Il seguente esempio di codice mostra come utilizzare `describe-alarm-history`.

### AWS CLI

Per recuperare la cronologia di un allarme

L'esempio seguente utilizza il `describe-alarm-history` comando per recuperare la cronologia dell'CloudWatch allarme Amazon denominato «myalarm»:

```
aws cloudwatch describe-alarm-history --alarm-name "myalarm" --history-item-type StateUpdate
```

Output:

```
{
  "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\"
  }
]
}
```

- Per API i dettagli, consulta AWS CLI Command [DescribeAlarmHistory](#) Reference.

## describe-alarms-for-metric

Il seguente esempio di codice mostra come utilizzare `describe-alarms-for-metric`.

### AWS CLI

Per visualizzare informazioni sugli allarmi associati a un parametro

L'esempio seguente utilizza il `describe-alarms-for-metric` comando per visualizzare informazioni su eventuali allarmi associati al EC2 CPUUtilization parametro Amazon e all'istanza con l'ID `i-0c986c72`.

```
aws cloudwatch describe-alarms-for-metric --metric-name CPUUtilization --
namespace AWS/EC2 --dimensions Name=InstanceId, Value=i-0c986c72
```

Output:

```
{
  \"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"
  }
]
}

```

- Per API i dettagli, vedere [DescribeAlarmsForMetric](#) in AWS CLI Command Reference.

## describe-alarms

Il seguente esempio di codice mostra come utilizzare `describe-alarms`.

### AWS CLI

Per elencare le informazioni di un allarme

L'esempio seguente utilizza il comando `describe-alarms` per fornire informazioni sull'allarme denominato "myalarm":

```
aws cloudwatch describe-alarms --alarm-names "myalarm"
```

Output:

```

{
  "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"
  }
]
}

```

- Per API i dettagli, vedere [DescribeAlarms](#) in AWS CLI Command Reference.

## disable-alarm-actions

Il seguente esempio di codice mostra come utilizzare `disable-alarm-actions`.

### AWS CLI

Per disattivare le operazioni su un allarme

L'esempio seguente utilizza il comando `disable-alarm-actions` per disabilitare tutte le operazioni per l'allarme denominato `myalarm`:

```
aws cloudwatch disable-alarm-actions --alarm-names myalarm
```

In caso di esito positivo, il comando torna al prompt.

- Per API i dettagli, vedere [DisableAlarmActions](#) in AWS CLI Command Reference.

## enable-alarm-actions

Il seguente esempio di codice mostra come utilizzare `enable-alarm-actions`.

### AWS CLI

Per abilitare tutte le operazioni relative a un allarme

L'esempio seguente utilizza il comando `enable-alarm-actions` per abilitare tutte le operazioni per l'allarme denominato `myalarm`.

```
aws cloudwatch enable-alarm-actions --alarm-names myalarm
```

In caso di esito positivo, il comando torna al prompt.

- Per API i dettagli, vedere [EnableAlarmActions](#) in AWS CLI Command Reference.

## get-metric-statistics

Il seguente esempio di codice mostra come utilizzare `get-metric-statistics`.

### AWS CLI

Per ottenere l'CPUutilizzo per istanza EC2

L'esempio seguente utilizza il `get-metric-statistics` comando per ottenere l'CPUutilizzo di un'EC2istanza con l'ID `i-abcdef`.

```
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
```

Output:

```
{
  "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"
}

```

### Specifica di più dimensioni

Nell'esempio seguente viene illustrato come specificare più dimensioni. Ogni dimensione è specificata come coppia nome/valore, con una virgola tra il nome e il valore. Più dimensioni sono separate da uno spazio. Se un unico parametro include più dimensioni, è necessario specificare un valore per ogni dimensione definita.

Per altri esempi di utilizzo del `get-metric-statistics` comando, consulta [Get Statistics for a Metric](#) nella 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
```

- Per API i dettagli, consulta [GetMetricStatistics AWS CLI Command Reference](#).

## list-metrics

Il seguente esempio di codice mostra come utilizzare `list-metrics`.

### AWS CLI

Per elencare le metriche per Amazon SNS

L'`list-metrics` esempio seguente mostra le metriche per Amazon SNS.

```
aws cloudwatch list-metrics \
  --namespace "AWS/SNS"
```

Output:

```
{
  "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"
  }
]
}

```

- Per API i dettagli, consulta [ListMetrics AWS CLI Command Reference](#).

## put-metric-alarm

Il seguente esempio di codice mostra come utilizzare `put-metric-alarm`.

### AWS CLI

Inviare un messaggio e-mail di Amazon Simple Notification Service quando CPU l'utilizzo supera il 70 per cento

L'esempio seguente utilizza il `put-metric-alarm` comando per inviare un messaggio e-mail di Amazon Simple Notification Service quando CPU l'utilizzo supera il 70 per cento:

```

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

```

In caso di esito positivo, il comando torna al prompt. Se esiste già un allarme con lo stesso nome, verrà sovrascritto dal nuovo allarme.

Per specificare più dimensioni

Nell'esempio seguente viene illustrato come specificare più dimensioni. Ogni dimensione è specificata come coppia nome/valore, con una virgola tra il nome e il valore. Più dimensioni sono separate da uno spazio:

```
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
```

- Per API i dettagli, consulta AWS CLI Command [PutMetricAlarm](#) Reference.

## put-metric-data

Il seguente esempio di codice mostra come utilizzare `put-metric-data`.

### AWS CLI

Per pubblicare una metrica personalizzata su Amazon CloudWatch

L'esempio seguente utilizza il `put-metric-data` comando per pubblicare una metrica personalizzata su Amazon CloudWatch:

```
aws cloudwatch put-metric-data --namespace "Usage Metrics" --metric-data file://  
metric.json
```

I valori della metrica stessa vengono memorizzati nel JSON file, `metric.json`

Ecco i contenuti del file:

```
[  
  {  
    "MetricName": "New Posts",  
    "Timestamp": "Wednesday, June 12, 2013 8:28:20 PM",  
    "Value": 0.50,  
    "Unit": "Count"  
  }  
]
```

Per ulteriori informazioni, consulta [Publishing Custom Metrics](#) nella Amazon CloudWatch Developer Guide.

Per specificare più dimensioni

Nell'esempio seguente viene illustrato come specificare più dimensioni. Ogni dimensione è specificata come coppia Name=Valore. Più dimensioni sono separate da una virgola:

```
aws cloudwatch put-metric-data --metric-name Buffers --  
namespace MyNameSpace --unit Bytes --value 231434333 --  
dimensions InstanceID=1-23456789,InstanceType=m1.small
```

- Per API i dettagli, consulta [PutMetricData AWS CLI](#) Command Reference.

## set-alarm-state

Il seguente esempio di codice mostra come utilizzare `set-alarm-state`.

### AWS CLI

Per modificare temporaneamente lo stato di un allarme

L'esempio seguente utilizza il `set-alarm-state` comando per modificare temporaneamente lo stato di un CloudWatch allarme Amazon denominato «myalarm» e impostarlo sullo ALARM stato a scopo di test:

```
aws cloudwatch set-alarm-state --alarm-name "myalarm" --state-value ALARM --state-  
reason "testing purposes"
```

In caso di esito positivo, il comando torna al prompt.

- Per API i dettagli, consulta [SetAlarmState AWS CLI](#) Command Reference.

## CloudWatch Esempi di log utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with CloudWatch Logs.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.



Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **create-log-group**

Il seguente esempio di codice mostra come utilizzare `create-log-group`.

#### AWS CLI

Il comando seguente crea un gruppo di log denominato `my-logs`:

```
aws logs create-log-group --log-group-name my-logs
```

- Per API i dettagli, vedere [CreateLogGroup](#) in AWS CLI Command Reference.

### **create-log-stream**

Il seguente esempio di codice mostra come utilizzare `create-log-stream`.

#### AWS CLI

Il comando seguente crea un flusso di log denominato `20150601` nel gruppo di log `my-logs`:

```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- Per API i dettagli, vedere [CreateLogStream](#) in AWS CLI Command Reference.

### **delete-log-group**

Il seguente esempio di codice mostra come utilizzare `delete-log-group`.

#### AWS CLI

Il comando seguente elimina un gruppo di log denominato `my-logs`:

```
aws logs delete-log-group --log-group-name my-Logs
```

- Per API i dettagli, vedere [DeleteLogGroup](#) in AWS CLI Command Reference.

## delete-log-stream

Il seguente esempio di codice mostra come utilizzare `delete-log-stream`.

### AWS CLI

Il comando seguente elimina un flusso di log denominato `20150531` da un gruppo di log denominato `my-logs`:

```
aws logs delete-log-stream --log-group-name my-logs --log-stream-name 20150531
```

- Per API i dettagli, vedere [DeleteLogStream](#) in AWS CLI Command Reference.

## delete-retention-policy

Il seguente esempio di codice mostra come utilizzare `delete-retention-policy`.

### AWS CLI

Il comando seguente rimuove la politica di conservazione che è stata precedentemente applicata a un gruppo di log denominato `my-logs`:

```
aws logs delete-retention-policy --log-group-name my-logs
```

- Per API i dettagli, vedere [DeleteRetentionPolicy](#) in AWS CLI Command Reference.

## describe-log-groups

Il seguente esempio di codice mostra come utilizzare `describe-log-groups`.

### AWS CLI

Il comando seguente descrive un gruppo di log denominato `my-logs`:

```
aws logs describe-log-groups --log-group-name-prefix my-logs
```

**Output:**

```
{
  "logGroups": [
    {
      "storedBytes": 0,
      "metricFilterCount": 0,
      "creationTime": 1433189500783,
      "logGroupName": "my-logs",
      "retentionInDays": 5,
      "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:*"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeLogGroups](#) in AWS CLI Command Reference.

**describe-log-streams**

Il seguente esempio di codice mostra come utilizzare `describe-log-streams`.

**AWS CLI**

Il comando seguente mostra tutti i flussi di log che iniziano con il prefisso `2015` nel gruppo di log: `my-logs`

```
aws logs describe-log-streams --log-group-name my-logs --log-stream-name-prefix 2015
```

**Output:**

```
{
  "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
  }
]
```

- Per API i dettagli, vedere [DescribeLogStreams](#) in AWS CLI Command Reference.

## get-log-events

Il seguente esempio di codice mostra come utilizzare `get-log-events`.

### AWS CLI

Il comando seguente recupera gli eventi di registro da un flusso di log denominato `20150601` nel gruppo `my-logs` di log:

```
aws logs get-log-events --log-group-name my-logs --log-stream-name 20150601
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [GetLogEvents](#) in AWS CLI Command Reference.

## put-log-events

Il seguente esempio di codice mostra come utilizzare `put-log-events`.

### AWS CLI

Il comando seguente inserisce gli eventi di registro in un flusso di log denominato `20150601` nel gruppo di log `my-logs`:

```
aws logs put-log-events --log-group-name my-logs --log-stream-name 20150601 --log-events file://events
```

Output:

```
{
  "nextSequenceToken": "49542672486831074009579604567656788214806863282469607346"
}
```

L'esempio precedente legge una JSON serie di eventi da un file denominato `events` nella directory corrente:

```
[
  {
    "timestamp": 1433190184356,
    "message": "Example Event 1"
  },
  {
    "timestamp": 1433190184358,
    "message": "Example Event 2"
  },
  {
    "timestamp": 1433190184360,
    "message": "Example Event 3"
  }
]
```

```
]
```

Ogni chiamata successiva richiede che il token di sequenza successivo fornito dalla chiamata precedente venga specificato con l'opzione `sequence token`:

```
aws logs put-log-events --log-group-name my-logs --log-  
stream-name 20150601 --log-events file://events2 --sequence-  
token "49542672486831074009579604567656788214806863282469607346"
```

Output:

```
{  
  "nextSequenceToken": "49542672486831074009579604567900991230369019956308219826"  
}
```

- Per API i dettagli, vedere [PutLogEvents](#) in AWS CLI Command Reference.

## put-retention-policy

Il seguente esempio di codice mostra come utilizzare `put-retention-policy`.

AWS CLI

Il comando seguente aggiunge una politica di conservazione di 5 giorni a un gruppo di log denominato `my-logs`:

```
aws logs put-retention-policy --log-group-name my-logs --retention-in-days 5
```

- Per API i dettagli, vedere [PutRetentionPolicy](#) in AWS CLI Command Reference.

## CloudWatch Esempi di monitoraggio della rete utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with CloudWatch Network Monitoring.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **create-monitor**

Il seguente esempio di codice mostra come utilizzare `create-monitor`.

## AWS CLI

Esempio 1: creare un monitor di rete con un periodo di aggregazione

L'`create-monitor` seguente crea un monitor denominato `Example_NetworkMonitor` con un valore `aggregationPeriod` impostato su 30 secondi. L'iniziale state del monitor sarà `INACTIVE` perché non vi sono sonde associate. Lo stato cambia `ACTIVE` solo quando vengono aggiunte le sonde. È possibile utilizzare i comandi [update-monitor](#) o [create-probe per aggiungere sonde](#) a questo monitor.

```
aws networkmonitor create-monitor \  
  --monitor-name Example_NetworkMonitor \  
  --aggregation-period 30
```

## Output:

```
{  
  "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/  
Example_NetworkMonitor",  
  "monitorName": "Example_NetworkMonitor",  
  "state": "INACTIVE",  
  "aggregationPeriod": 30,  
  "tags": {}  
}
```

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

Esempio 2: per creare un monitor di rete con una sonda, utilizzare TCP e includere tag

L'create-monitor esempio seguente crea un monitor denominato `Example_NetworkMonitor`. Il comando crea anche una sonda che utilizza il ICMP protocollo e include tag. Poiché nella richiesta `aggregationPeriod` viene passato `no`, i 60 secondi sono impostati come valore predefinito. Il state monitor con la sonda rimarrà attivo `PENDING` fino allo spegnimento del monitor. `ACTIVE` Questa operazione potrebbe richiedere alcuni minuti, dopodiché state cambierà e potrai iniziare a `ACTIVE` visualizzare le CloudWatch metriche.

```
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
```

Output:

```
{
  "monitorArn": "arn:aws:networkmonitor:region111122223333:monitor/
Example_NetworkMonitor",
  "monitorName": "Example_NetworkMonitor",
  "state": "PENDING",
  "aggregationPeriod": 60,
  "tags": {
    "Monitor": "Monitor1"
  }
}
```

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

Esempio 3: per creare un monitor di rete con una sonda, utilizzare ICMP e includere tag

L'create-monitor esempio seguente crea un monitor denominato `Example_NetworkMonitor` con un `aggregationPeriod` di 30 secondi. Il comando crea anche una sonda che utilizza il ICMP protocollo e include tag. Poiché nella richiesta `aggregationPeriod` viene passato `no`, i 60 secondi sono impostati come valore predefinito. Il state monitor con la sonda rimarrà attivo `PENDING` fino allo spegnimento del monitor. `ACTIVE`



Questa operazione potrebbe richiedere alcuni minuti, dopodiché state cambierà e potrai iniziare a ACTIVE visualizzare le CloudWatch metriche.

```
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
```

Output:

```
{  
  "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/  
Example_NetworkMonitor",  
  "monitorName": "Example_NetworkMonitor",  
  "state": "PENDING",  
  "aggregationPeriod": 30,  
  "tags": {  
    "Monitor": "Monitor1"  
  }  
}
```

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

- Per API i dettagli, consulta [CreateMonitor AWS CLI Command Reference](#).

## create-probe

Il seguente esempio di codice mostra come utilizzare `create-probe`.

### AWS CLI

Esempio 1: creare una sonda che la utilizzi TCP e aggiungerla a un monitor di rete

L'`create-probe`esempio seguente crea una sonda che utilizza TCP protocol e aggiunge la sonda a un monitor denominato. `Example_NetworkMonitor` Una volta creato, il state monitor con la sonda rimarrà in funzione PENDING finché il monitor non lo sarà. ACTIVE Questa operazione potrebbe richiedere alcuni minuti, dopodiché lo stato cambierà e potrai iniziare a ACTIVE visualizzare le CloudWatch metriche.

```
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}
```

Output:

```
{
  "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"
  }
}
```

Esempio 2: creare una sonda che utilizza probe using ICMP e aggiungerla a un monitor di rete

L'create-probeesempio seguente crea una sonda che utilizza ICMP protocol e aggiunge la sonda a un monitor denominato. Example\_NetworkMonitor Una volta creato, il state monitor con la sonda rimarrà in funzione PENDING finché il monitor non lo sarà. ACTIVE Questa operazione potrebbe richiedere alcuni minuti, dopodiché lo stato cambierà e potrai iniziare a ACTIVE visualizzare le CloudWatch metriche.

```
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}
```

Output:

```
{
  "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"
}
}
```

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

- Per API i dettagli, consulta [CreateProbe AWS CLI Command Reference](#).

## delete-monitor

Il seguente esempio di codice mostra come utilizzare `delete-monitor`.

### AWS CLI

Per eliminare un monitor

L'esempio seguente elimina un monitor denominato `Example_NetworkMonitor`.

```
aws networkmonitor delete-monitor \
  --monitor-name Example_NetworkMonitor
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

- Per API i dettagli, consulta [DeleteMonitor AWS CLI Command Reference](#).

## delete-probe

Il seguente esempio di codice mostra come utilizzare `delete-probe`.

## AWS CLI

Per eliminare una sonda

L'`delete-probe` seguente elimina una sonda con l'ID `probe-12345` da un monitor di rete denominato `Example_NetworkMonitor`

```
aws networkmonitor delete-probe \  
  --monitor-name Example_NetworkMonitor \  
  --probe-id probe-12345
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

- Per API i dettagli, consulta [DeleteProbe AWS CLI Command Reference](#).

## `get-monitor`

Il seguente esempio di codice mostra come utilizzare `get-monitor`.

## AWS CLI

Per ottenere informazioni sul monitor

L'`get-monitor` seguente ottiene informazioni su un monitor denominato `Example_NetworkMonitor`.

```
aws networkmonitor get-monitor \  
  --monitor-name Example_NetworkMonitor
```

Output:

```
{  
  "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"
}
```

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

- Per API i dettagli, consulta [GetMonitor AWS CLI Command Reference](#).

## get-probe

Il seguente esempio di codice mostra come utilizzare `get-probe`.

### AWS CLI

Per visualizzare i dettagli della sonda

L'`get-probe` esempio seguente restituisce i dettagli su una sonda con `probeID` `probe-12345` il nome associato a un monitor. `Example_NetworkMonitor`

```
aws networkmonitor get-probe \
  --monitor-name Example_NetworkMonitor \
  --probe-id probe-12345
```

Output:

```
{
  "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"
  }
}
```

```
}  
}
```

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

- Per API i dettagli, consulta [GetProbe AWS CLI](#) Command Reference.

## list-monitors

Il seguente esempio di codice mostra come utilizzare `list-monitors`.

### AWS CLI

Esempio 1: Per elencare tutti i monitor (monitor singolo)

L'`list-monitors` esempio seguente restituisce un elenco di un solo monitor. Il monitor state è `ACTIVE` e ha una durata `aggregationPeriod` di 60 secondi.

```
aws networkmonitor list-monitors
```

Output:

```
{  
  "monitors": [{  
    "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor",  
    "monitorName": "Example_NetworkMonitor",  
    "state": "ACTIVE",  
    "aggregationPeriod": 60,  
    "tags": {  
      "Monitor": "Monitor1"  
    }  
  }  
]  
}
```

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

Esempio 2: per elencare tutti i monitor (più monitor)

L'`list-monitors` seguente restituisce un elenco di tre monitor. La funzione `state` di un monitor è `ACTIVE` e la generazione di CloudWatch metriche. Gli stati degli altri due monitor sono `INACTIVE` e non CloudWatch generano metriche. Tutti e tre i monitor utilizzano `60` secondi `aggregationPeriod`.

```
aws networkmonitor list-monitors
```

Output:

```
{
  "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": {}
    }
  ]
}
```

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

- Per API i dettagli, consulta [ListMonitors AWS CLI Command Reference](#).

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una risorsa

L'`list-tags-for-resource` esempio seguente restituisce un elenco dei tag per un monitor denominato `Example_NetworkMonitor`.

```
aws networkmonitor list-tags-for-resource \
  --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor
```

Output:

```
{
  "tags": {
    "Environment": "Dev",
    "Application": "PetStore"
  }
}
```

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare una risorsa

L'`tag-resource` esempio seguente contrassegna un monitor denominato `Example_NetworkMonitor` con `Application=PetStore` tag `Environment=Dev` e.



```
aws networkmonitor tag-resource \  
  --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor \  
  --tags Environment=Dev,Application=PetStore
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

- Per API i dettagli, consulta [TagResource AWS CLI](#) Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una risorsa

L'esempio seguente rimuove un tag-key parametro con la coppia chiave-valore di `Environment Application` dalla sua associazione a un monitor denominato `Example_NetworkMonitor`

```
aws networkmonitor untag-resource \  
  --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor \  
  --tag-keys Environment Application
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

- Per API i dettagli, consulta [UntagResource AWS CLI](#) Command Reference.

## update-monitor

Il seguente esempio di codice mostra come utilizzare `update-monitor`.

## AWS CLI

Per aggiornare un monitor

L'update-monitor esempio seguente modifica il valore di un monitor aggregationPeriod da 60 secondi a 30 secondi.

```
aws networkmonitor update-monitor \  
  --monitor-name Example_NetworkMonitor \  
  --aggregation-period 30
```

Output:

```
{  
  "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/  
Example_NetworkMonitor",  
  "monitorName": "Example_NetworkMonitor",  
  "state": "PENDING",  
  "aggregationPeriod": 30,  
  "tags": {  
    "Monitor": "Monitor1"  
  }  
}
```

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

- Per API i dettagli, consulta [UpdateMonitor AWS CLI Command Reference](#).

## update-probe

Il seguente esempio di codice mostra come utilizzare update-probe.

## AWS CLI

Per aggiornare una sonda

L'update-probe esempio seguente aggiorna l'indirizzo destination IP originale di una sonda e aggiorna anche il packetSize. 60

```
aws networkmonitor update-probe \  
  --destination-ip 10.0.0.1 \  
  --packet-size 60
```

```
--monitor-name Example_NetworkMonitor \  
--probe-id probe-12345 \  
--destination 10.0.0.150 \  
--packet-size 60
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [How CloudWatch Network Monitor Works](#) nella Amazon CloudWatch User Guide.

- Per API i dettagli, consulta [UpdateProbe AWS CLI](#) Command Reference.

## CodeArtifact esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with CodeArtifact.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **associate-external-connection**

Il seguente esempio di codice mostra come utilizzare `associate-external-connection`.

#### AWS CLI

Per aggiungere una connessione esterna a un repository

L'esempio seguente aggiunge una connessione esterna a `npmjs.com` a un repository denominato `test-repo`.

```
aws codeartifact associate-external-connection \  
  --repository test-repo \  
  --domain test-domain \  
  --external-connection public:npmjs
```

Output:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Aggiungere](#) una connessione esterna nella Guida per l'utente.AWS CodeArtifact

- Per API i dettagli, vedere [AssociateExternalConnection](#)in AWS CLI Command Reference.

## copy-package-versions

Il seguente esempio di codice mostra come utilizzare `copy-package-versions`.

### AWS CLI

Per copiare le versioni dei pacchetti da un repository all'altro

Quanto segue `copy-package-versions` sposta le versioni 4.0.0 e 5.0.0 di un pacchetto chiamato `test-package` da `my-repo` a `test-repo`.

```
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"]'
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

[Per ulteriori informazioni, consulta Copiare i pacchetti tra i repository nella Guida per l'utente.AWS CodeArtifact](#)

- Per API i dettagli, vedere [CopyPackageVersions](#) in AWS CLI Command Reference.

## create-domain

Il seguente esempio di codice mostra come utilizzare `create-domain`.

### AWS CLI

Per creare un dominio

L'`create-domain` esempio seguente crea un dominio denominato `test-domain`.

```
aws codeartifact create-domain \  
  --domain test-domain
```

Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, consulta [Creare un dominio nella Guida](#) per l'AWS CodeArtifact utente.

- Per API i dettagli, vedere [CreateDomain](#) in AWS CLI Command Reference.

## create-repository

Il seguente esempio di codice mostra come utilizzare `create-repository`.

## AWS CLI

Per creare un repository

L'`create-repository` seguente crea un repository denominato `test-repo` all'interno di un dominio denominato `test-domain`.

```
aws codeartifact create-repository \  
  --domain test-domain \  
  --domain-owner 111122223333 \  
  --repository test-repo \  
  --description "This is a test repository."
```

Output:

```
{  
  "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": []  
  }  
}
```

Per ulteriori informazioni, consulta [Creare un](#) dominio nella Guida per l'utente.AWS CodeArtifact

- Per API i dettagli, vedere [CreateRepository](#) in AWS CLI Command Reference.

## **delete-domain-permissions-policy**

Il seguente esempio di codice mostra come utilizzare `delete-domain-permissions-policy`.

## AWS CLI

Per eliminare il documento relativo alla politica delle autorizzazioni da un dominio

L'`delete-domain-permissions-policy` seguente elimina la politica di autorizzazione da un dominio denominato `test-domain`.

```
aws codeartifact delete-domain-permissions-policy \  
  --domain test-domain
```

Output:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Eliminare una politica di dominio nella Guida](#) per l'AWS CodeArtifact utente.

- Per API i dettagli, vedere [DeleteDomainPermissionsPolicy](#) in AWS CLI Command Reference.

## delete-domain

Il seguente esempio di codice mostra come utilizzare `delete-domain`.

### AWS CLI

Per eliminare un dominio

L'`delete-domain` seguente elimina un dominio denominato `test-domain`.

```
aws codeartifact delete-domain \  
  --domain test-domain
```



```
--domain test-domain
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Eliminare un dominio](#) nella Guida per l'AWS CodeArtifact utente.

- Per API i dettagli, vedere [DeleteDomain](#) in AWS CLI Command Reference.

## delete-package-versions

Il seguente esempio di codice mostra come utilizzare `delete-package-versions`.

AWS CLI

Per eliminare le versioni dei pacchetti

L'esempio seguente elimina la versione 4.0.0 di un pacchetto denominato `test-package`.

```
aws codeartifact delete-package-versions \
  --domain test-domain \
  --repo test-repo \
  --format npm \
  --package test-package \
  --versions 4.0.0
```

Output:

```
{
  "successfulVersions": {
    "4.0.0": {
      "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
      "status": "Deleted"
    }
  },
  "failedVersions": {}
}
```

Per ulteriori informazioni, consulta [Eliminare una versione del pacchetto](#) nella Guida per l'utente.AWS CodeArtifact

- Per API i dettagli, vedere [DeletePackageVersions](#)in AWS CLI Command Reference.

## delete-repository-permissions-policy

Il seguente esempio di codice mostra come utilizzare `delete-repository-permissions-policy`.

### AWS CLI

Per eliminare una politica di autorizzazioni da un repository

L'`delete-repository-permissions-policy`esempio seguente elimina la politica di autorizzazione da un repository denominato `test-repo`.

```
aws codeartifact delete-repository-permissions-policy \
  --domain test-domain \
  --repository test-repo
```

Output:

```
{
  "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": "*"
}
]
}

```

Per ulteriori informazioni, consulta [Eliminare una politica](#) nella Guida per l'utente AWS CodeArtifact.

- Per API i dettagli, vedere [DeleteRepositoryPermissionsPolicy](#) in AWS CLI Command Reference.

## delete-repository

Il seguente esempio di codice mostra come utilizzare `delete-repository`.

### AWS CLI

Per eliminare un repository

L'esempio seguente elimina un repository denominato `test-repo` in un dominio denominato `test-domain`.

```

aws codeartifact delete-repository \
  --domain test-domain \
  --repository test-repo

```

Output:

```

{
  "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": []
  }
}
```

Per ulteriori informazioni, vedere [Eliminare un repository](#) nella Guida per l'AWS CodeArtifact utente.

- Per API i dettagli, vedere [DeleteRepository](#) in AWS CLI Command Reference.

## describe-domain

Il seguente esempio di codice mostra come utilizzare `describe-domain`.

### AWS CLI

Per ottenere informazioni su un dominio

L'`describe-domain` esempio seguente restituisce un `DomainDescription` oggetto per un dominio denominato `test-domain`.

```
aws codeartifact describe-domain \
  --domain test-domain
```

Output:

```
{
  "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"
```

```
}  
}
```

Per ulteriori informazioni, vedere [Panoramica del dominio nella Guida](#) per l'AWS CodeArtifact utente.

- Per API i dettagli, vedere [DescribeDomain](#) in AWS CLI Command Reference.

## describe-repository

Il seguente esempio di codice mostra come utilizzare `describe-repository`.

### AWS CLI

Per ottenere informazioni su un repository

L'`describe-repository` esempio seguente restituisce un `RepositoryDescription` oggetto per un repository denominato `test-repo`.

```
aws codeartifact describe-repository \  
  --domain test-domain \  
  --repository test-repo
```

Output:

```
{  
  "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": []  
  }  
}
```

Per ulteriori informazioni, consulta [Creare un dominio](#) nella Guida per l'utente AWS CodeArtifact.

- Per API i dettagli, vedere [DescribeRepository](#) in AWS CLI Command Reference.

## disassociate-external-connection

Il seguente esempio di codice mostra come utilizzare `disassociate-external-connection`.

### AWS CLI

Per rimuovere una connessione esterna da un repository

L'esempio seguente rimuove una connessione esterna a `npmjs.com` da un repository denominato `test-repo`.

```
aws codeartifact disassociate-external-connection \  
  --repository test-repo \  
  --domain test-domain \  
  --external-connection public:npmjs
```

Output:

```
{  
  "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": []  
  }  
}
```

Per ulteriori informazioni, consulta [Rimuovere](#) una connessione esterna nella Guida per l'utente AWS CodeArtifact.

- Per API i dettagli, vedere [DisassociateExternalConnection](#) in AWS CLI Command Reference.

## dispose-package-versions

Il seguente esempio di codice mostra come utilizzare `dispose-package-versions`.

### AWS CLI

Per eliminare gli asset di una versione del pacchetto e impostarne lo stato su `Disposed`

L'`dispose-package-versions`esempio seguente elimina gli asset della versione `test-package 4.0.0` e ne imposta lo stato su `Disposed`.

```
aws codeartifact dispose-package-versions \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --versions 4.0.0
```

Output:

```
{  
  "successfulVersions": {  
    "4.0.0": {  
      "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",  
      "status": "Disposed"  
    }  
  },  
  "failedVersions": {}  
}
```

Per ulteriori informazioni, consulta [Lavorare con i pacchetti](#) nella Guida per l'utente. CodeArtifact AWS CodeArtifact

- Per API i dettagli, vedere [DisposePackageVersions](#)in AWS CLI Command Reference.

## get-authorization-token

Il seguente esempio di codice mostra come utilizzare `get-authorization-token`.

AWS CLI

Per ottenere un token di autorizzazione

L'`get-authorization-token`esempio seguente recupera un token di CodeArtifact autorizzazione.

```
aws codeartifact get-authorization-token \  
  --domain test-domain \  
  --query authorizationToken \  
  --output text
```

```
--output text
```

### Output:

This command will return the authorization token. You can store the output in an environment variable when calling the command.

Per ulteriori informazioni, consulta [Configura pip senza il comando login nella Guida](#) per l'AWS CodeArtifact utente.

- Per API i dettagli, vedere [GetAuthorizationToken](#) in AWS CLI Command Reference.

## get-domain-permissions-policy

Il seguente esempio di codice mostra come utilizzare `get-domain-permissions-policy`.

### AWS CLI

Per ottenere il documento relativo alla politica delle autorizzazioni per un dominio

L'`get-domain-permissions-policy` esempio seguente ottiene la politica di autorizzazione allegata a un dominio denominato `test-domain`.

```
aws codeartifact get-domain-permissions-policy \  
  --domain test-domain
```

### Output:

```
{  
  "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"
        }
    ]
}

```

Per ulteriori informazioni, consulta [Leggere una politica di dominio nella Guida](#) per l'AWS CodeArtifact utente.

- Per API i dettagli, vedere [GetDomainPermissionsPolicy](#) in AWS CLI Command Reference.

## get-package-version-asset

Il seguente esempio di codice mostra come utilizzare `get-package-version-asset`.

### AWS CLI

Per ottenere una risorsa da una versione del pacchetto

L'`get-package-version-asset` esempio seguente recupera la `package.tgz` risorsa per la versione 4.0.0 di un pacchetto npm denominato `test-package`.

```

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

```

Output:

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="
}

```

Per ulteriori informazioni, vedete [Elencare gli asset della versione del pacchetto nella Guida](#) per l'utente.AWS CodeArtifact

- Per API i dettagli, consultate [GetPackageVersionAsset AWS CLI Command Reference](#).

## get-package-version-readme

Il seguente esempio di codice mostra come utilizzare `get-package-version-readme`.

### AWS CLI

Per ottenere il file readme della versione del pacchetto

L'esempio seguente recupera il file readme per la versione 4.0.0 di un pacchetto npm denominato `test-package`.

```
aws codeartifact get-package-version-readme \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --package-version 4.0.0
```

Output:

```
{  
  "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="  
}
```

Per ulteriori informazioni, consulta [Visualizza il file readme della versione del pacchetto](#) nella Guida per l'utente.AWS CodeArtifact

- Per API i dettagli, vedere [GetPackageVersionReadme](#) in AWS CLI Command Reference.

## get-repository-endpoint

Il seguente esempio di codice mostra come utilizzare `get-repository-endpoint`.

## AWS CLI

Per ottenere l'endpoint di un repository URL

L'`get-repository-endpoint` seguente restituisce l'endpoint npm per il repository `test-repo`.

```
aws codeartifact get-repository-endpoint \  
  --domain test-domain \  
  --repository test-repo \  
  --format npm
```

Output:

```
{  
  "repositoryEndpoint": "https://test-domain-111122223333.d.codeartifact.us-  
west-2.amazonaws.com/npm/test-repo/"  
}
```

Per ulteriori informazioni, consulta [Connect to a repository](#) nella Guida per l'AWS CodeArtifact utente.

- Per API i dettagli, vedere [GetRepositoryEndpoint](#) in AWS CLI Command Reference.

## `get-repository-permissions-policy`

Il seguente esempio di codice mostra come utilizzare `get-repository-permissions-policy`.

## AWS CLI

Per ottenere il documento relativo alla politica delle autorizzazioni per un repository

L'`get-repository-permissions-policy` seguente ottiene la politica di autorizzazione allegata a un repository denominato `test-repo`.

```
aws codeartifact get-repository-permissions-policy \  
  --domain test-domain \  
  --repository test-repo
```

Output:

```
{
```

```

"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": "*"
  }
]
}

```

Per ulteriori informazioni, consulta [Leggere una politica](#) nella Guida per l'utente.AWS CodeArtifact

- Per API i dettagli, vedere [GetRepositoryPermissionsPolicy](#) in AWS CLI Command Reference.

## list-domains

Il seguente esempio di codice mostra come utilizzare `list-domains`.

### AWS CLI

Per elencare i domini

L'`list-domains`esempio seguente restituisce un riepilogo di tutti i domini di proprietà dell' AWS account che effettua la chiamata.

```
aws codeartifact list-domains
```

Output:

```
{
```

```
"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"  
  }  
]
```

Per ulteriori informazioni, vedere [Lavorare con i domini CodeArtifact nella Guida](#) per l'AWS CodeArtifact utente.

- Per API i dettagli, vedere [ListDomains](#) in AWS CLI Command Reference.

## list-package-version-assets

Il seguente esempio di codice mostra come utilizzare `list-package-version-assets`.

### AWS CLI

Per visualizzare gli asset di una versione del pacchetto

L'`list-package-version-assets` esempio seguente recupera gli asset per la versione 4.0.0 di un pacchetto npm denominato `test-package`.

```
aws codeartifact list-package-version-assets \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --package-version 4.0.0
```

Output:

```
{
  "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"
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta [Elenca gli asset della versione del pacchetto nella Guida](#) per l'utente.AWS CodeArtifact

- Per API i dettagli, consultate [ListPackageVersionAssets AWS CLI](#) Command Reference.

## list-package-version-dependencies

Il seguente esempio di codice mostra come utilizzare `list-package-version-dependencies`.

### AWS CLI

Per visualizzare le dipendenze della versione di un pacchetto

L'`list-package-version-dependencies` esempio seguente recupera le dipendenze per la versione 4.0.0 di un pacchetto npm denominato `test-package`.

```
aws codeartifact list-package-version-dependencies \
  --domain test-domain \
  --repo test-repo \
  --format npm \
  --package test-package \
```

```
--package-version 4.0.0
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Visualizzare e aggiornare i dettagli e le dipendenze della versione del pacchetto nella Guida](#) per l'utente.AWS CodeArtifact

- Per API i dettagli, vedere [ListPackageVersionDependencies](#)in AWS CLI Command Reference.

## list-package-versions

Il seguente esempio di codice mostra come utilizzare `list-package-versions`.

AWS CLI

Per elencare le versioni di un pacchetto

L'`list-package-versions`esempio seguente restituisce un elenco di versioni di pacchetto per un pacchetto denominato `kind-of`.

```
aws codeartifact list-package-versions \  
  --package kind-of \  
  --
```

```
--domain test-domain \  
--repository test-repo \  
--format npm
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Elenco le versioni del pacchetto](#) nella Guida AWS CodeArtifact per l'utente.

- Per API i dettagli, vedere [ListPackageVersions](#) in AWS CLI Command Reference.



## list-packages

Il seguente esempio di codice mostra come utilizzare `list-packages`.

### AWS CLI

Per elencare i pacchetti in un repository

L'esempio seguente elenca i pacchetti in un repository denominato `test-repo` in un dominio denominato `test-domain`

```
aws codeartifact list-packages \
  --domain test-domain \
  --repository test-repo
```

Output:

```
{
  "packages": [
    {
      "format": "npm",
      "package": "lodash"
    }
    {
      "format": "python",
      "package": "test-package"
    }
  ]
}
```

Per ulteriori informazioni, vedete [Elencare i nomi dei pacchetti](#) nella Guida per l'AWS CodeArtifact utente.

- Per API i dettagli, vedere [ListPackages](#) in AWS CLI Command Reference.

## list-repositories-in-domain

Il seguente esempio di codice mostra come utilizzare `list-repositories-in-domain`.

### AWS CLI

Per elencare i repository in un dominio

L'`list-repositories-in-domain` seguente restituisce un riepilogo di tutti i repository nel dominio `test-domain`.

```
aws codeartifact list-repositories-in-domain \  
  --domain test-domain
```

Output:

```
{  
  "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."  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [List repositories](#) nella User Guide.AWS CodeArtifact

- Per API i dettagli, vedere [ListRepositoriesInDomain](#) in AWS CLI Command Reference.

## **list-repositories**

Il seguente esempio di codice mostra come utilizzare `list-repositories`.

AWS CLI

Per elencare i repository

L' `list-repositories` esempio seguente restituisce un riepilogo di tutti gli archivi del dominio di proprietà dell' AWS account che effettua la chiamata.

```
aws codeartifact list-repositories
```

Output:

```
{
  "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."
    }
  ]
}
```

Per ulteriori informazioni, consulta [List repositories](#) nella AWS CodeArtifact User Guide.

- Per API i dettagli, vedere [ListRepositories](#) in AWS CLI Command Reference.

## login

Il seguente esempio di codice mostra come utilizzare `login`.

### AWS CLI

Per configurare l'autenticazione al tuo repository con il comando `login`

L'esempio seguente configura il gestore di pacchetti `npm` con un repository denominato `test-repo` in un dominio chiamato `test-domain`.

```
aws codeartifact login \  
  --domain test-domain \  
  --repository test-repo \  
  --tool npm
```

Output:

```
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
```

Per ulteriori informazioni, consulta [Getting started with the](#) nella Guida per l'utente. AWS CLI AWS CodeArtifact

- Per API i dettagli, consulta [Login](#) in AWS CLI Command Reference.

## put-domain-permissions-policy

Il seguente esempio di codice mostra come utilizzare `put-domain-permissions-policy`.

### AWS CLI

Per allegare una politica di autorizzazioni a un dominio

L'esempio seguente allega una politica di autorizzazione definita nel file `policy.json` a un dominio denominato `test-domain`.

```
aws codeartifact put-domain-permissions-policy \  
  --domain test-domain \  
  --policy-document file://PATH/TO/policy.json
```

**Output:**

```
{
  "policy": {
    "resourceArn": "arn:aws:codeartifact:region-id:111122223333:domain/test-
domain",
    "document": "{ ...policy document content...}",
    "revision": "MQlyyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxxxx="
  }
}
```

Per ulteriori informazioni, consulta [Impostare una politica di dominio](#) nella Guida per l'utente AWS CodeArtifact

- Per API i dettagli, vedere [PutDomainPermissionsPolicy](#) in AWS CLI Command Reference.

**put-repository-permissions-policy**

Il seguente esempio di codice mostra come utilizzare `put-repository-permissions-policy`.

**AWS CLI**

Per allegare una politica di autorizzazioni a un repository

L'esempio seguente allega una politica di autorizzazione definita nel file `policy.json` a un repository denominato `test-repo`.

```
aws codeartifact put-repository-permissions-policy \
  --domain test-domain \
  --repository test-repo \
  --policy-document file://PATH/T0/policy.json
```

**Output:**

```
{
  "policy": {
    "resourceArn": "arn:aws:codeartifact:region-id:111122223333:repository/test-
domain/test-repo",
    "document": "{ ...policy document content...}",
    "revision": "MQlyyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxxxx="
  }
}
```

```
}
```

Per ulteriori informazioni, consulta [Impostare una politica nella guida per l'utente.AWS CodeArtifact](#)

- Per API i dettagli, vedere [PutRepositoryPermissionsPolicy](#) in AWS CLI Command Reference.

## update-package-versions-status

Il seguente esempio di codice mostra come utilizzare `update-package-versions-status`.

### AWS CLI

Per aggiornare lo stato della versione del pacchetto

L'`update-package-versions-status`esempio seguente aggiorna lo stato della versione 4.0.0 del pacchetto `test-package` su `Archived`.

```
aws codeartifact update-package-versions-status \  
  --domain test-domain \  
  --repo test-repo \  
  --format npm \  
  --package test-package \  
  --versions 4.0.0 \  
  --target-status Archived
```

Output:

```
{  
  "successfulVersions": {  
    "4.0.0": {  
      "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",  
      "status": "Archived"  
    }  
  },  
  "failedVersions": {}  
}
```

Per ulteriori informazioni, consulta [Aggiornare lo stato della versione del pacchetto nella Guida](#) per l'utente.AWS CodeArtifact

- Per API i dettagli, vedere [UpdatePackageVersionsStatus](#) in AWS CLI Command Reference.

## update-repository

Il seguente esempio di codice mostra come utilizzare `update-repository`.

### AWS CLI

Per aggiornare un repository

L'esempio seguente aggiorna la descrizione di un repository denominato `test-repo` in un dominio denominato `test-domain` in «questa è una descrizione aggiornata».

```
aws codeartifact update-repository \
  --domain test-domain \
  --repository test-repo \
  --description "this is an updated description"
```

Output:

```
{
  "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": []
  }
}
```

Per ulteriori informazioni, consulta [Visualizzare o modificare la configurazione di un repository](#) nella Guida per l'utente AWS CodeArtifact.

- Per API i dettagli, vedere [UpdateRepository](#) in AWS CLI Command Reference.

## CodeBuild esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with CodeBuild.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **batch-delete-builds**

Il seguente esempio di codice mostra come utilizzare `batch-delete-builds`.

AWS CLI

Per eliminare le build in AWS CodeBuild.

L'`batch-delete-builds`esempio seguente elimina le build in base a CodeBuild quanto specificato. IDs

```
aws codebuild batch-delete-builds --ids my-build-project-one:a1b2c3d4-5678-9012-abcd-11111EXAMPLE my-build-project-two:a1b2c3d4-5678-9012-abcd-22222EXAMPLE
```

Output:

```
{
  "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"
  ]
}
```



Per ulteriori informazioni, vedete [Delete Builds \(AWS CLI\)](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [BatchDeleteBuilds](#) in AWS CLI Command Reference.

## batch-get-build-batches

Il seguente esempio di codice mostra come utilizzare `batch-get-build-batches`.

### AWS CLI

Per visualizzare i dettagli delle build in AWS CodeBuild.

L'`batch-get-build-batches` esempio seguente ottiene informazioni sui batch di compilazione CodeBuild con quanto specificato. IDs

```
aws codebuild batch-get-build-batches \  
  --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE
```

Output:

```
{  
  "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": []
      }
    }
  ]
}
```

```
    },
    {
      "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": []
}

```

Per ulteriori informazioni, consulta Batch builds in AWS CodeBuild < <https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html> > \_\_ nella Guida per l'utente.AWS CodeBuild

- Per i API dettagli, vedere in Command Reference. [BatchGetBuildBatches](#) AWS CLI

## batch-get-builds

Il seguente esempio di codice mostra come utilizzare `batch-get-builds`.

### AWS CLI

Per visualizzare i dettagli delle build in AWS CodeBuild.

L'`batch-get-builds`esempio seguente ottiene informazioni sulle compilazioni CodeBuild con quanto specificato. IDs

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE
```

Output:

```

{
  "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"
    }
]
}

```

Per ulteriori informazioni, vedete [View Build Details \(AWS CLI\)](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [BatchGetBuilds](#) in AWS CLI Command Reference.

## batch-get-projects

Il seguente esempio di codice mostra come utilizzare `batch-get-projects`.

### AWS CLI

Per ottenere un elenco dei nomi dei progetti di AWS CodeBuild compilazione.

L'`batch-get-projects` esempio seguente ottiene un elenco di progetti di CodeBuild compilazione specificati per nome.

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-project2 my-other-demo-project
```

Nell'output seguente, l'`projectsNotFound` array elenca tutti i nomi di progetto di build che sono stati specificati, ma non trovati. L'array `projects` elenca i dettagli per ogni progetto di compilazione di cui sono state trovate informazioni.

```

{
  "projectsNotFound": [],
  "projects": [
    {

```



```
"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": []
}
]
```

Per ulteriori informazioni, consulta [View a Build Project's Details \(AWS CLI\)](#) nella Guida AWS CodeBuild per l'utente.



- Per API i dettagli, vedere [BatchGetProjects](#) in AWS CLI Command Reference.

## batch-get-report-groups

Il seguente esempio di codice mostra come utilizzare `batch-get-report-groups`.

### AWS CLI

Per ottenere informazioni su uno o più gruppi di report in AWS CodeBuild.

L'esempio seguente recupera le informazioni sul gruppo di report con il valore specificato ARN.

```
aws codebuild batch-get-report-groups \
  --report-group-arns arn:aws:codebuild:<region-ID>:<user-ID>:report-group/
<report-group-name>
```

Output:

```
{
  "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": []
}
```

Per ulteriori informazioni, vedere [Lavorare con i gruppi di report](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [BatchGetReportGroups](#) in AWS CLI Command Reference.

## batch-get-reports

Il seguente esempio di codice mostra come utilizzare `batch-get-reports`.

### AWS CLI

Per ottenere informazioni su uno o più report in AWS CodeBuild.

L'esempio seguente recupera le informazioni sui report con il valore specificato ARNs.

```
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>
```

Output:

```
{
  "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": []
}

```

Per ulteriori informazioni, vedere [Utilizzo dei report](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [BatchGetReports](#) in AWS CLI Command Reference.

## create-project

Il seguente esempio di codice mostra come utilizzare `create-project`.

### AWS CLI

Esempio 1: creare un progetto di AWS CodeBuild compilazione

L'esempio seguente crea un progetto di CodeBuild compilazione utilizzando i file sorgente di un bucket S3

```
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"
```

Output:

```
{
  "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
  }
}

```

Esempio 2: creare un progetto di AWS CodeBuild compilazione utilizzando un file JSON di input per i parametri

L'`create-project` seguente crea un progetto di CodeBuild compilazione passando tutti i parametri richiesti in un file JSON di input. Crea il modello del file di input eseguendo il comando solo con `--generate-cli-skeleton parameter`.

```
aws codebuild create-project --cli-input-json file://create-project.json
```

Il JSON file di input `create-project.json` contiene il seguente contenuto:

```

{
  "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"
}

```

### Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Create a Build Project \(AWS CLI\)](#) nella Guida AWS CodeBuild per l'utente.

- Per API i dettagli, consultate [CreateProject AWS CLI Command Reference](#).

## create-report-group

Il seguente esempio di codice mostra come utilizzare `create-report-group`.

### AWS CLI

Per creare un gruppo di report in AWS CodeBuild.

L'`create-report-group` seguente crea un nuovo gruppo di report.

```
aws codebuild create-report-group \  
  --cli-input-json file://create-report-group-source.json
```

Contenuto di `create-report-group-source.json`:

```
{  
  "name": "cli-created-report-group",  
  "type": "TEST",  
  "exportConfig": {  
    "exportConfigType": "S3",  
    "s3Destination": {  
      "bucket": "my-s3-bucket",  
      "path": "",  
      "packaging": "ZIP",  
      "encryptionDisabled": true  
    }  
  }  
}
```

Output:

```
{  
  "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
  }
}

```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di report nella Guida](#) per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [CreateReportGroup](#) in AWS CLI Command Reference.

## create-webhook

Il seguente esempio di codice mostra come utilizzare `create-webhook`.

### AWS CLI

Per creare filtri webhook per un progetto AWS CodeBuild

L'`create-webhook` seguente crea un webhook per un CodeBuild progetto denominato `my-project` che ha due gruppi di filtri. Il primo gruppo di filtri specifica le richieste pull create, aggiornate o riaperte su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/master$` e riferimenti head che corrispondono a `^refs/heads/myBranch$`. Il secondo gruppo di filtri specifica le richieste push sui rami con nomi di riferimento Git che non corrispondono all'espressione `^refs/heads/myBranch$` regolare.

```

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}]]"

```



## Output:

```
{
  "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
        }
      ]
    ]
  }
}
```

Per ulteriori informazioni, consulta [Filter GitHub Webhook Events \(SDK\) nella Guida](#) per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [CreateWebhookin](#) AWS CLI Command Reference.

## delete-build-batch

Il seguente esempio di codice mostra come utilizzare `delete-build-batch`.

### AWS CLI

Per eliminare un batch, compila AWS CodeBuild.

L'`delete-build-batch` esempio seguente elimina la build batch specificata.

```
aws codebuild delete-build-batch \  
  --id <project-name>:<batch-ID>
```

Output:

```
{  
  "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": []  
}
```

Per ulteriori informazioni, consulta [Batch builds AWS CodeBuild in](#) the AWS CodeBuild User Guide.

- Per API i dettagli, vedere [DeleteBuildBatchin](#) AWS CLI Command Reference.

## delete-project

Il seguente esempio di codice mostra come utilizzare `delete-project`.

## AWS CLI

Per eliminare un progetto di AWS CodeBuild compilazione

L'`delete-project` seguente elimina il progetto di CodeBuild compilazione specificato.

```
aws codebuild delete-project --name my-project
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Delete a Build Project \(AWS CLI\)](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [DeleteProject](#) in AWS CLI Command Reference.

## `delete-report-group`

Il seguente esempio di codice mostra come utilizzare `delete-report-group`.

### AWS CLI

Per eliminare un report si raggruppa in AWS CodeBuild.

L'`delete-report-group` seguente elimina il gruppo di report con quanto specificato ARN.

```
aws codebuild delete-report-group \  
  --arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-group-name>
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Utilizzo dei gruppi di report](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [DeleteReportGroup](#) in AWS CLI Command Reference.

## `delete-report`

Il seguente esempio di codice mostra come utilizzare `delete-report`.

### AWS CLI

Per eliminare un rapporto in AWS CodeBuild.

L'`delete-report` seguente elimina il report specificato.

```
aws codebuild delete-report \  
  --arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Utilizzo dei report](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [DeleteReport](#) in AWS CLI Command Reference.

## `delete-source-credentials`

Il seguente esempio di codice mostra come utilizzare `delete-source-credentials`.

### AWS CLI

Per disconnettersi da un provider di origine e rimuoverne i token di accesso.

L'`delete-source-credentials` seguente si disconnette da un provider di origine e ne rimuove i token. Le credenziali ARN di origine utilizzate per connettersi al provider di origine determinano le credenziali di origine.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

Output:

```
{  
  "arn": "arn:aws:codebuild:your-region:your-account-id:token/your-server-type"  
}
```

Per ulteriori informazioni, consulta [Connect Source Providers with Access Tokens \(CLI\)](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [DeleteSourceCredentials](#) in AWS CLI Command Reference.

## `delete-webhook`

Il seguente esempio di codice mostra come utilizzare `delete-webhook`.

## AWS CLI

Per eliminare un filtro webhook da un progetto AWS CodeBuild

L'`delete-webhook` seguente elimina un webhook dal progetto specificato. CodeBuild

```
aws codebuild delete-webhook --project-name my-project
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Stop Running Builds Automaticamente \(AWS CLI\) nella Guida per l'AWS CodeBuild utente](#).

- Per API i dettagli, vedere [DeleteWebhook](#) in AWS CLI Command Reference.

## `describe-code-coverages`

Il seguente esempio di codice mostra come utilizzare `describe-code-coverages`.

## AWS CLI

Per ottenere informazioni dettagliate sulla copertura del codice, i risultati dei test in AWS CodeBuild.

L'`describe-code-coverages` seguente ottiene informazioni sui risultati del test di copertura del codice nel rapporto specificato.

```
aws codebuild describe-code-coverages \  
  --report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

Output:

```
{  
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta i [report sulla copertura del codice](#) nella Guida AWS CodeBuild per l'utente.

- Per API i dettagli, vedere [DescribeCodeCoverages](#) in AWS CLI Command Reference.

## describe-test-cases

Il seguente esempio di codice mostra come utilizzare `describe-test-cases`.

### AWS CLI

Per ottenere informazioni dettagliate sui casi di test in AWS CodeBuild.

L'esempio seguente ottiene informazioni sui casi di test nel rapporto specificato.

```

aws codebuild describe-test-cases \
  --report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-
name>:<report-ID>

```

Output:

```

{
  "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"
    }
  ]
}

```

Per ulteriori informazioni, vedere [Working with test reporting AWS CodeBuild nella Guida per l'AWS CodeBuild utente](#).

- Per API i dettagli, vedere [DescribeTestCases](#) in AWS CLI Command Reference.

## import-source-credentials

Il seguente esempio di codice mostra come utilizzare `import-source-credentials`.

### AWS CLI

Connect un AWS CodeBuild utente a un provider di origine importando le credenziali per il provider di origine.

L'`import-source-credentials` esempio seguente importa un token per un repository Bitbucket che utilizza `BASIC_AUTH` per il suo tipo di autenticazione.

```
aws codebuild import-source-credentials --server-type BITBUCKET --auth-  
type BASIC_AUTH --token my-Bitbucket-password --username my-Bitbucket-username
```

Output:

```
{  
  "arn": "arn:aws:codebuild:us-west-2:123456789012:token/bitbucket"  
}
```

Per ulteriori informazioni, consulta [Connect Source Providers with Access Tokens \(CLI\)](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [ImportSourceCredentials](#) in AWS CLI Command Reference.

## **invalidate-project-cache**

Il seguente esempio di codice mostra come utilizzare `invalidate-project-cache`.

AWS CLI

Per reimpostare la cache per un progetto di AWS CodeBuild compilazione.

L'`invalidate-project-cache` esempio seguente reimposta la cache per il CodeBuild progetto specificato.

```
aws codebuild invalidate-project-cache --project-name my-project
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Build Caching in CodeBuild nella Guida](#) per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [InvalidateProjectCache](#) in AWS CLI Command Reference.

## **list-build-batches-for-project**

Il seguente esempio di codice mostra come utilizzare `list-build-batches-for-project`.



## AWS CLI

Per elencare le build in batch per uno specifico progetto di compilazione in AWS CodeBuild.

L'`list-build-batches-for-project` seguente elenca le build CodeBuild in batch per il progetto specificato.

```
aws codebuild list-build-batches-for-project \
  --project-name "<project-name>"
```

Output:

```
{
  "ids": [
    "<project-name>:<batch-ID>",
    "<project-name>:<batch-ID>"
  ]
}
```

Per ulteriori informazioni, consulta [Batch builds AWS CodeBuild in](#) the AWS CodeBuild User Guide.

- Per API i dettagli, vedere [ListBuildBatchesForProject](#) in AWS CLI Command Reference.

## `list-build-batches`

Il seguente esempio di codice mostra come utilizzare `list-build-batches`.

## AWS CLI

Per elencare le build in AWS CodeBuild batch.

L'`list-build-batches` seguente elenca le build CodeBuild in batch per l'account corrente.

```
aws codebuild list-build-batches
```

Output:

```
{
```

```
"ids": [  
  "<project-name>:<batch-ID>",  
  "<project-name>:<batch-ID>"  
]  
}
```

Per ulteriori informazioni, consulta Batch builds in AWS CodeBuild < <https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html> > \_\_ nella Guida per l'utente.AWS CodeBuild

- Per i API dettagli, vedere in Command Reference. [ListBuildBatches](#) AWS CLI

## list-builds-for-project

Il seguente esempio di codice mostra come utilizzare `list-builds-for-project`.

### AWS CLI

Per visualizzare un elenco di build per un progetto di AWS CodeBuild compilazione.

L'`list-builds-for-project` esempio seguente elenca la build IDs in ordine decrescente per il progetto di CodeBuild compilazione specificato.

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order DESCENDING
```

Output:

```
{  
  "ids": [  
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-1111example",  
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-2222example",  
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-3333example",  
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-4444example",  
    "codebuild-demo-project:1a2b3c4d-5678-90ab-cdef-5555example"  
  ]  
}
```

Per ulteriori informazioni, vedete [Visualizzare un elenco di build IDs for a Build Project \(AWS CLI\)](#) nella Guida per l'AWS CodeBuild utente

- Per API i dettagli, vedere [ListBuildsForProject](#) in AWS CLI Command Reference.

## list-builds

Il seguente esempio di codice mostra come utilizzare `list-builds`.

### AWS CLI

Per ottenere un elenco di AWS CodeBuild build. IDs

L'`list-builds` esempio seguente ottiene un elenco CodeBuild IDs ordinato in ordine crescente.

```
aws codebuild list-builds --sort-order ASCENDING
```

L'output include un `nextToken` valore che indica che c'è più output disponibile.

```
{
  "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"
  ]
}
```

Esegui nuovamente questo comando e fornisci il `nextToken` valore della risposta precedente come parametro per ottenere la parte successiva dell'output. Ripeti finché non ricevi più un `nextToken` valore nella risposta.

```
aws codebuild list-builds --sort-order ASCENDING --next-
token 4AEA6u7J...The full token has been omitted for brevity...MzY20A==
```

Parte successiva dell'output:

```
{
  "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"
  ]
}
```

Per ulteriori informazioni, vedete [View a List of Build IDs \(AWS CLI\)](#) nella Guida AWS CodeBuild per l'utente

- Per API i dettagli, vedere [ListBuilds](#) in AWS CLI Command Reference.

## list-curated-environment-images

Il seguente esempio di codice mostra come utilizzare `list-curated-environment-images`.

### AWS CLI

Per ottenere un elenco di immagini Docker gestite da AWS CodeBuild che puoi utilizzare per le tue build.

L'`list-curated-environment-images` esempio seguente elenca le immagini Docker gestite da CodeBuild che possono essere utilizzate per le build. :

```
aws codebuild list-curated-environment-images
```

Output:

```
{
  "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 ...
  ]
}

```

Per ulteriori informazioni, consulta [le immagini Docker fornite da CodeBuild nella Guida per l'AWS CodeBuild utente](#)

- Per API i dettagli, vedere [ListCuratedEnvironmentImages](#) in AWS CLI Command Reference.

## list-projects

Il seguente esempio di codice mostra come utilizzare `list-projects`.

### AWS CLI

Per ottenere un elenco dei nomi dei progetti di AWS CodeBuild compilazione.

L'`list-projects` esempio seguente ottiene un elenco di progetti di CodeBuild compilazione ordinati per nome in ordine crescente.

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

L'output include un `nextToken` valore che indica che c'è più output disponibile.

```

{
  "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"
  ]
}

```

Esegui nuovamente questo comando e fornisci il `nextToken` valore della risposta precedente come parametro per ottenere la parte successiva dell'output. Ripeti finché non ricevi più un `nextToken` valore nella risposta.

```
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"  
  ]  
}
```

Per ulteriori informazioni, consulta [Visualizza un elenco di nomi di progetti di compilazione \(AWS CLI\)](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [ListProjects](#) in AWS CLI Command Reference.

## list-report-groups

Il seguente esempio di codice mostra come utilizzare `list-report-groups`.

### AWS CLI

Per ottenere un elenco del gruppo di report ARNs in AWS CodeBuild.

L'`list-report-groups` esempio seguente recupera il gruppo di rapporti ARNs per l'account nella regione.

```
aws codebuild list-report-groups
```

Output:

```
{  
  "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"  
  ]  
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di report](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [ListReportGroups](#) in AWS CLI Command Reference.

## list-reports-for-report-group

Il seguente esempio di codice mostra come utilizzare `list-reports-for-report-group`.

### AWS CLI

Per ottenere un elenco dei report in un report, raggruppa in AWS CodeBuild.

L'`list-report-for-report-group` esempio seguente recupera i report nel gruppo di rapporti specificato per l'account nella regione.

```
aws codebuild list-reports-for-report-group \  
  --report-group-arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-  
group-name>
```

Output:

```
{  
  "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"  
  ]  
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di report](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [ListReportsForReportGroup](#) in AWS CLI Command Reference.

## list-reports

Il seguente esempio di codice mostra come utilizzare `list-reports`.

## AWS CLI

Per ottenere un elenco dei report relativi all'account corrente in AWS CodeBuild.

L'`list-reports`esempio seguente recupera ARNs i report per l'account corrente.

```
aws codebuild list-reports
```

Output:

```
{
  "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>"
  ]
}
```

Per ulteriori informazioni, vedere [Utilizzo dei report](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [ListReports](#)in AWS CLI Command Reference.

## `list-shared-projects`

Il seguente esempio di codice mostra come utilizzare `list-shared-projects`.

## AWS CLI

In cui elencare il progetto condiviso AWS CodeBuild.

L'`list-shared-projects`esempio seguente elenca i progetti CodeBuild condivisi disponibili per l'account corrente.

```
aws codebuild list-shared-projects
```

Output:

```
{
  "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>"  
  ]  
}
```

Per ulteriori informazioni, vedere [Lavorare con progetti condivisi](#) nella Guida AWS CodeBuild per l'utente.

- Per API i dettagli, vedere [ListSharedProjects](#) in AWS CLI Command Reference.

## list-shared-report-groups

Il seguente esempio di codice mostra come utilizzare `list-shared-report-groups`.

### AWS CLI

Per ottenere un elenco del gruppo di report condiviso ARNs in AWS CodeBuild.

L'`list-shared-report-groups` esempio seguente recupera il gruppo di report ARNs per l'account nella regione.

```
aws codebuild list-shared-report-groups
```

Output:

```
{  
  "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"  
  ]  
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di report](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [ListSharedReportGroups](#) in AWS CLI Command Reference.

## list-source-credentials

Il seguente esempio di codice mostra come utilizzare `list-source-credentials`.

## AWS CLI

Per visualizzare un elenco di `sourceCredentialsObjects`

L'`list-source-credentials` seguente elenca i token per un AWS account collegato a un account Bitbucket e a un account. GitHub Ogni `sourceCredentialsInfos` oggetto nella risposta contiene informazioni sulle credenziali della fonte connessa.

```
aws codebuild list-source-credentials
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Connect Source Providers with Access Tokens \(CLI\)](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [ListSourceCredentials](#) in AWS CLI Command Reference.

## retry-build-batch

Il seguente esempio di codice mostra come utilizzare `retry-build-batch`.

### AWS CLI

Per riprovare la compilazione di un batch non riuscito. AWS CodeBuild

L'`retry-build-batch` seguente riavvia la compilazione in batch specificata.

```
aws codebuild retry-build-batch \  
  --id <project-name>:<batch-ID>
```

## Output:

```
{  
  "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"
      }
    }
  ]
}

```

Per ulteriori informazioni, consulta [Batch builds AWS CodeBuild in](#) the AWS CodeBuild User Guide.

- Per API i dettagli, vedere [RetryBuildBatch](#) in AWS CLI Command Reference.

## retry-build

Il seguente esempio di codice mostra come utilizzare `retry-build`.

### AWS CLI

Per riprovare una compilazione non riuscita. AWS CodeBuild

L'`retry-build` seguente riavvia la build specificata.

```
aws codebuild retry-build \  
  --id <project-name>:<build-ID>
```

Output:

```
{  
  "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"
}
}

```

Per ulteriori informazioni, consulta [Batch builds AWS CodeBuild in the AWS CodeBuild User Guide](#).

- Per API i dettagli, vedere [RetryBuild](#) in AWS CLI Command Reference.



## start-build-batch

Il seguente esempio di codice mostra come utilizzare `start-build-batch`.

### AWS CLI

Per avviare una compilazione in batch AWS CodeBuild.

L'esempio seguente avvia una compilazione in batch del progetto specificato.

```
aws codebuild start-build-batch \  
  --project-name <project-name>
```

Output:

```
{  
  "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
    }
  }
}

```

Per ulteriori informazioni, consulta [Batch builds AWS CodeBuild in](#) the AWS CodeBuild User Guide.

- Per API i dettagli, vedere [StartBuildBatch](#) in AWS CLI Command Reference.

## start-build

Il seguente esempio di codice mostra come utilizzare `start-build`.

## AWS CLI

Per iniziare a eseguire la build di un progetto di AWS CodeBuild compilazione.

L'`start-build` seguente avvia una compilazione per il CodeBuild progetto specificato. La build sostituisce sia l'impostazione del progetto per il numero di minuti in cui la build può essere messa in coda prima che scada sia le impostazioni degli artefatti del progetto.

```
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}
```

Output:

```
{  
  "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"
}
}

```

Per ulteriori informazioni, consulta [Run a Build \(\)AWS CLI](#) nella Guida per l'utente.AWS CodeBuild

- Per API i dettagli, vedere [StartBuild](#)in AWS CLI Command Reference.

## stop-build-batch

Il seguente esempio di codice mostra come utilizzarestop-build-batch.

### AWS CLI

Per interrompere un batch in corso, compila. AWS CodeBuild

L'`stop-build-batches` seguente interrompe la compilazione in batch specificata.

```
aws codebuild stop-build-batch \  
  --id <project-name>:<batch-ID>
```

Output:

```
{  
  "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"
    }
}
]
}
}

```

Per ulteriori informazioni, consulta [Batch builds AWS CodeBuild in the AWS CodeBuild User Guide](#).

- Per API i dettagli, vedere [StopBuildBatch](#) in AWS CLI Command Reference.

## stop-build

Il seguente esempio di codice mostra come utilizzare `stop-build`.

### AWS CLI

Per interrompere la compilazione di un progetto di AWS CodeBuild compilazione.

L'`stop-build` esempio seguente interrompe la CodeBuild compilazione specificata.

```
aws codebuild stop-build --id my-demo-project:12345678-a1b2-c3d4-e5f6-11111EXAMPLE
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, vedete [Stop a Build \(AWS CLI\)](#) nella Guida AWS CodeBuild per l'utente.

- Per API i dettagli, vedere [StopBuild](#) in AWS CLI Command Reference.

## update-project

Il seguente esempio di codice mostra come utilizzare `update-project`.

### AWS CLI

Per modificare le impostazioni di un progetto di AWS CodeBuild compilazione.

L'`update-project` esempio seguente modifica le impostazioni del progetto di CodeBuild compilazione specificato denominato `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"
```

L'output mostra le impostazioni aggiornate.

```
{  
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Modifica delle impostazioni di un progetto di compilazione \(AWS CLI\)](#) nella Guida AWS CodeBuild per l'utente

- Per API i dettagli, consulta [UpdateProject AWS CLI](#) Command Reference.

## update-report-group

Il seguente esempio di codice mostra come utilizzare `update-report-group`.

### AWS CLI

Per aggiornare un gruppo di report in AWS CodeBuild.

L'`update-report-group`esempio seguente modifica il tipo di esportazione del gruppo di report in «NO\_EXPORT».

```
aws codebuild update-report-group \
```

```
--arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-report-  
group \  
--export-config="exportConfigType=NO_EXPORT"
```

Output:

```
{  
  "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": []  
  }  
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di report](#) nella Guida per l'AWS CodeBuild utente.

- Per API i dettagli, vedere [UpdateReportGroup](#) in AWS CLI Command Reference.

## update-webhook

Il seguente esempio di codice mostra come utilizzare `update-webhook`.

### AWS CLI

Per aggiornare il webhook di un progetto AWS CodeBuild

L'esempio seguente aggiorna un webhook per il CodeBuild progetto specificato con due gruppi di filtri. Il `--rotate-secret` parametro specifica che la chiave segreta del progetto deve ruotare ogni volta che una modifica al codice attiva una compilazione. Il primo gruppo di filtri specifica le richieste pull create, aggiornate o riaperte su rami con nomi di riferimento Git che corrispondono all'espressione regolare `^refs/heads/master$` e riferimenti head che corrispondono a `^refs/heads/myBranch$`. Il secondo gruppo di filtri specifica le richieste push sui rami con nomi di riferimento Git che non corrispondono all'espressione `^refs/heads/myBranch$` regolare.

```
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}]]"
```

Output:

```
{
  "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  
  }  
}
```

Per maggiori informazioni, consulta [Change a Build Project \(AWS CLI\)](#) nella Guida per l'AWS CodeBuild utente

- Per API i dettagli, consulta [UpdateWebhook AWS CLI](#) Command Reference.

## CodeCommit esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with CodeCommit.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **associate-approval-rule-template-with-repository**

Il seguente esempio di codice mostra come utilizzare `associate-approval-rule-template-with-repository`.

AWS CLI

Per associare un modello di regola di approvazione a un repository

L'esempio seguente associa il modello di regola di approvazione specificato a un archivio denominato `MyDemoRepo`

```
aws codecommit associate-approval-rule-template-with-repository \  
  --repository-name MyDemoRepo \  
  \
```



```
--approval-rule-template-name 2-approver-rule-for-main
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Associare un modello di regola di approvazione a un repository](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [AssociateApprovalRuleTemplateWithRepository](#) in AWS CLI Command Reference.

## batch-associate-approval-rule-template-with-repositories

Il seguente esempio di codice mostra come utilizzare `batch-associate-approval-rule-template-with-repositories`.

### AWS CLI

Associare un modello di regola di approvazione a più repository in un'unica operazione

L'esempio seguente associa il modello di regola di approvazione specificato ai repository denominati `MyDemoRepo` e `MyOtherDemoRepo`.

Nota: i modelli di regole di approvazione sono specifici della AWS regione in cui vengono creati. Possono essere associati solo ai repository di quella AWS regione.

```
aws codecommit batch-associate-approval-rule-template-with-repositories \  
  --repository-names MyDemoRepo, MyOtherDemoRepo \  
  --approval-rule-template-name 2-approver-rule-for-main
```

Output:

```
{  
  "associatedRepositoryNames": [  
    "MyDemoRepo",  
    "MyOtherDemoRepo"  
  ],  
  "errors": []  
}
```

Per ulteriori informazioni, consulta [Associare un modello di regola di approvazione a un repository](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [BatchAssociateApprovalRuleTemplateWithRepositories](#) in AWS CLI Command Reference.

## batch-describe-merge-conflicts

Il seguente esempio di codice mostra come utilizzare `batch-describe-merge-conflicts`.

### AWS CLI

Per ottenere informazioni sui conflitti di unione in tutti i file o in un sottoinsieme di file in un'unione tra due specificatori di commit

L'esempio seguente determina i conflitti di unione per l'unione di un ramo di origine denominato `feature-randomizationfeature` con un ramo di destinazione denominato `main` utilizzando la strategia in un repository denominato `THREE_WAY_MERGE` `MyDemoRepo`

```
aws codecommit batch-describe-merge-conflicts \  
  --source-commit-specifier feature-randomizationfeature \  
  --destination-commit-specifier main \  
  --merge-option THREE_WAY_MERGE \  
  --repository-name MyDemoRepo
```

Output:

```
{  
  "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"
}

```

Per ulteriori informazioni, consulta [Resolve Conflicts in a Pull Request nella Guida](#) per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [BatchDescribeMergeConflicts](#) in AWS CLI Command Reference.

## batch-disassociate-approval-rule-template-from-repositories

Il seguente esempio di codice mostra come utilizzare `batch-disassociate-approval-rule-template-from-repositories`.

### AWS CLI

Per dissociare un modello di regola di approvazione da più repository in un'unica operazione

L'esempio seguente dissocia il modello di regola di approvazione specificato dai repository denominati `MyDemoRepo` e `MyOtherDemoRepo`.

```
aws codecommit batch-disassociate-approval-rule-template-from-repositories \
  --repository-names MyDemoRepo, MyOtherDemoRepo \
  --approval-rule-template-name 1-approval-rule-for-all pull requests
```

Output:

```
{
  "disassociatedRepositoryNames": [
    "MyDemoRepo",
    "MyOtherDemoRepo"
  ],
  "errors": []
}
```

Per ulteriori informazioni, consulta [Dissociare un modello di regola di approvazione](#) nella Guida per l'utente AWS CodeCommit.

- Per API i dettagli, vedere [BatchDisassociateApprovalRuleTemplateFromRepositories](#) in AWS CLI Command Reference.

## batch-get-commits

Il seguente esempio di codice mostra come utilizzare `batch-get-commits`.

### AWS CLI

Per visualizzare informazioni su più commit

L'batch-get-commits seguente visualizza i dettagli sui commit specificati.

```
aws codecommit batch-get-commits \
  --repository-name MyDemoRepo \
  --commit-ids 317f8570EXAMPLE 4c925148EXAMPLE
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, vedere [View Commit Details](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [BatchGetCommits](#) in AWS CLI Command Reference.

## batch-get-repositories

Il seguente esempio di codice mostra come utilizzare `batch-get-repositories`.

### AWS CLI

Per visualizzare i dettagli su più repository

Questo esempio mostra i dettagli su più AWS CodeCommit repository.

```

aws codecommit batch-get-repositories \
  --repository-names MyDemoRepo MyOtherDemoRepo

```

Output:

```

{
  "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": []
}

```

- Per API i dettagli, vedere [BatchGetRepositories](#) in AWS CLI Command Reference.

## create-approval-rule-template

Il seguente esempio di codice mostra come utilizzare `create-approval-rule-template`.

### AWS CLI

Per creare un modello di regola di approvazione

L'esempio seguente crea un modello di regola di approvazione denominato `2-approver-rule-for-main`. The template requires two users who assume the role of `CodeCommitReview` per approvare qualsiasi pull request prima che possa essere unita al main ramo.

```

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/*"]}]}

```

**Output:**

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Creare un modello di regola di approvazione nella Guida](#) per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [CreateApprovalRuleTemplate](#) in AWS CLI Command Reference.

**create-branch**

Il seguente esempio di codice mostra come utilizzare `create-branch`.

**AWS CLI**

Per creare un ramo

Questo esempio crea un ramo in un AWS CodeCommit repository. Questo comando restituisce un output solo se ci sono errori.

Comando:

```
aws codecommit create-branch --repository-name MyDemoRepo --branch-name MyNewBranch
--commit-id 317f8570EXAMPLE
```

**Output:**



None.

- Per API i dettagli, vedere [CreateBranch](#) in AWS CLI Command Reference.

## create-commit

Il seguente esempio di codice mostra come utilizzare `create-commit`.

### AWS CLI

Per creare un commit

L'`create-commit` esempio seguente dimostra come creare un commit iniziale per un repository che aggiunge un `readme.md` file a un repository denominato `MyDemoRepo` nel ramo `main`

```
aws codecommit create-commit \  
  --repository-name MyDemoRepo \  
  --branch-name main \  
  --put-files "filePath=readme.md,fileContent='Welcome to our team repository.'"
```

Output:

```
{  
  "filesAdded": [  
    {  
      "blobId": "5e1c309d-EXAMPLE",  
      "absolutePath": "readme.md",  
      "fileMode": "NORMAL"  
    }  
  ],  
  "commitId": "4df8b524-EXAMPLE",  
  "treeId": "55b57003-EXAMPLE",  
  "filesDeleted": [],  
  "filesUpdated": []  
}
```

Per ulteriori informazioni, consulta [Create a Commit AWS CodeCommit nella Guida](#) per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [CreateCommit](#) in AWS CLI Command Reference.

## create-pull-request-approval-rule

Il seguente esempio di codice mostra come utilizzare `create-pull-request-approval-rule`.

### AWS CLI

Per creare una regola di approvazione per una pull request

L'esempio seguente crea una regola di approvazione denominata `Require two approved approvers` per la pull request specificata. La regola specifica che sono necessarie due approvazioni da un pool di approvazioni. Il pool include tutti gli utenti che accedono CodeCommit assumendo il ruolo di membro dell'account `CodeCommitReview.123456789012` AWS. Include anche un IAM utente o un utente federato denominato `Nikhil_Jayashankar` dallo stesso AWS account.

```
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/*\"]}]}"
```

Output:

```
{
  "approvalRule": {
    "approvalRuleName": "Require two approved approvers",
    "lastModifiedDate": 1570752871.932,
    "ruleContentSha256": "7c44e6ebEXAMPLE",
    "creationDate": 1570752871.932,
    "approvalRuleId": "aac33506-EXAMPLE",
    "approvalRuleContent": "{\"Version\": \"2018-11-08\", \"Statements\":
    [{\"Type\": \"Approvers\", \"NumberOfApprovalsNeeded\": 2, \"ApprovalPoolMembers
    \": [\"CodeCommitApprovers:123456789012:Nikhil_Jayashankar\",
    \"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
    "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major"
  }
}
```

Per ulteriori informazioni, consulta [Creare una regola di approvazione](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [CreatePullRequestApprovalRule](#) in AWS CLI Command Reference.

## create-pull-request

Il seguente esempio di codice mostra come utilizzare `create-pull-request`.

### AWS CLI

Per creare una pull request

L'`create-pull-request` esempio seguente crea una richiesta pull denominata «Pronunciation difficulty analyzer» con una descrizione di «Please review these changes by Tuesday» che ha come destinazione il ramo di origine «jane-branch» e deve essere unita al ramo predefinito «main» in un repository denominato ". 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
```

Output:

```
{
  "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
  }
}

```

- Per i API dettagli, vedere in Command Reference. [CreatePullRequest](#) AWS CLI

## create-repository

Il seguente esempio di codice mostra come utilizzare `create-repository`.

### AWS CLI

Per creare un repository

Questo esempio crea un repository e lo associa all'account dell'utente. AWS

Comando:

```
aws codecommit create-repository --repository-name MyDemoRepo --repository-
description "My demonstration repository"
```

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [CreateRepository](#) in AWS CLI Command Reference.

## create-unreferenced-merge-commit

Il seguente esempio di codice mostra come utilizzare `create-unreferenced-merge-commit`.

### AWS CLI

Per creare un commit non referenziato che rappresenti il risultato della fusione di due specificatori di commit

L'esempio seguente crea un commit che rappresenta i risultati di un'unione tra un ramo di origine denominato `bugfix-1234` con un ramo di destinazione denominato `main` utilizzando la strategia `THREE _ WAY _ MERGE` in un repository denominato `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."
```

**Output:**

```
{
  "commitId": "4f178133EXAMPLE",
  "treeId": "389765daEXAMPLE"
}
```

Per ulteriori informazioni, consulta [Resolve Conflicts in a Pull Request](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [CreateUnreferencedMergeCommit](#) in AWS CLI Command Reference.

**credential-helper**

Il seguente esempio di codice mostra come utilizzare `credential-helper`.

**AWS CLI**

Per configurare l'helper per le credenziali incluso in with AWS CLI AWS CodeCommit

L'`credential-helper` utilità non è progettata per essere chiamata direttamente da AWS CLI. È invece pensata per essere utilizzata come parametro con il `git config` comando per configurare il computer locale. Consente a Git di utilizzare HTTPS una versione firmata crittograficamente delle credenziali IAM utente o del ruolo dell'EC2 istanza Amazon ogni volta che Git deve autenticarsi per interagire con AWS i repository. CodeCommit

```
git config --global credential.helper '!aws codecommit credential-helper $@'
git config --global credential.UseHttpPath true
```

**Output:**

```
[credential]
  helper = !aws codecommit credential-helper $@
  UseHttpPath = true
```

Per ulteriori informazioni, consulta Configurazione per l' AWS CodeCommit utilizzo di altri metodi nella Guida per l'utente. AWS CodeCommit Esamina attentamente il contenuto, quindi segui le procedure in uno dei seguenti argomenti: Per HTTPS connessioni su Linux, macOS o Unix o Per HTTPS connessioni su Windows nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [CredentialHelper](#) in AWS CLI Command Reference.

## delete-approval-rule-template

Il seguente esempio di codice mostra come utilizzare `delete-approval-rule-template`.

### AWS CLI

Per eliminare un modello di regola di approvazione

L'esempio seguente elimina il modello di regola di approvazione specificato.

```
aws codecommit delete-approval-rule-template \
  --approval-rule-template-name 1-approver-for-all-pull-requests
```

Output:

```
{
  "approvalRuleTemplateId": "41de97b7-EXAMPLE"
}
```

Per ulteriori informazioni, vedere [Eliminare un modello di regola di approvazione](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [DeleteApprovalRuleTemplate](#) in AWS CLI Command Reference.

## delete-branch

Il seguente esempio di codice mostra come utilizzare `delete-branch`.

### AWS CLI

Per eliminare un ramo

Questo esempio mostra come eliminare un ramo in un AWS CodeCommit repository.

Comando:

```
aws codecommit delete-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

**Output:**

```
{
  "branch": {
    "commitId": "317f8570EXAMPLE",
    "branchName": "MyNewBranch"
  }
}
```

- Per API i dettagli, vedere [DeleteBranch](#) in AWS CLI Command Reference.

**delete-comment-content**

Il seguente esempio di codice mostra come utilizzare `delete-comment-content`.

**AWS CLI**

Per eliminare il contenuto di un commento

È possibile eliminare i contenuti di un commento solo se è stato creato personalmente. Questo esempio dimostra come eliminare il contenuto di un commento con l'ID generato dal sistema di `ff30b348EXAMPLEb9aa670f`

```
aws codecommit delete-comment-content \
  --comment-id ff30b348EXAMPLEb9aa670f
```

**Output:**

```
{
  "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
    }
  }
}
```



```
}  
}
```

- Per API i dettagli, vedere [DeleteCommentContent](#) in AWS CLI Command Reference.

## delete-file

Il seguente esempio di codice mostra come utilizzare `delete-file`.

### AWS CLI

Per eliminare un file

L'esempio seguente mostra come eliminare un file denominato `README.md` da un ramo denominato `main` con un ID di commit più recente di `c5709475EXAMPLE` in un repository denominato `MyDemoRepo`.

```
aws codecommit delete-file \  
  --repository-name MyDemoRepo \  
  --branch-name main \  
  --file-path README.md \  
  --parent-commit-id c5709475EXAMPLE
```

Output:

```
{  
  "blobId": "559b44fEXAMPLE",  
  "commitId": "353cf655EXAMPLE",  
  "filePath": "README.md",  
  "treeId": "6bc824cEXAMPLE"  
}
```

Per ulteriori informazioni, consulta [Modificare o eliminare un file AWS CodeCommit nella guida AWS CodeCommit API](#) di riferimento.

- Per API i dettagli, vedere [DeleteFile](#) in AWS CLI Command Reference.

## delete-pull-request-approval-rule

Il seguente esempio di codice mostra come utilizzare `delete-pull-request-approval-rule`.

## AWS CLI

Per eliminare una regola di approvazione per una pull request

L'`delete-pull-request-approval-rule` seguente elimina la regola di approvazione denominata `My Approval Rule` per la pull request specificata.

```
aws codecommit delete-pull-request-approval-rule \
  --approval-rule-name "My Approval Rule" \
  --pull-request-id 15
```

Output:

```
{
  "approvalRuleId": "077d8e8a8-EXAMPLE"
}
```

Per ulteriori informazioni, consulta [Modificare o eliminare una regola di approvazione](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [DeletePullRequestApprovalRule](#) in AWS CLI Command Reference.

## **delete-repository**

Il seguente esempio di codice mostra come utilizzare `delete-repository`.

## AWS CLI

Per eliminare un repository

Questo esempio mostra come eliminare un AWS CodeCommit repository.

Comando:

```
aws codecommit delete-repository --repository-name MyDemoRepo
```

Output:

```
{
  "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"
}
```

```
}
```

- Per API i dettagli, vedere [DeleteRepository](#) in AWS CLI Command Reference.

## describe-merge-conflicts

Il seguente esempio di codice mostra come utilizzare `describe-merge-conflicts`.

### AWS CLI

Per ottenere informazioni dettagliate sui conflitti di fusione

L'`describe-merge-conflicts` esempio seguente determina i conflitti di unione per un file denominato `readme.md` nel ramo di origine e nel ramo di destinazione specificati utilizzando la strategia `THREE_WAY_MERGE`.

```
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
```

Output:

```
{  
  "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"
}

```

Per ulteriori informazioni, vedere [Resolve Conflicts in a Pull Request](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [DescribeMergeConflicts](#) in AWS CLI Command Reference.

## describe-pull-request-events

Il seguente esempio di codice mostra come utilizzare `describe-pull-request-events`.

## AWS CLI

Per visualizzare gli eventi in una pull request

L'`describe-pull-request-events` seguente recupera gli eventi per una richiesta pull con l'ID '8'.

```
aws codecommit describe-pull-request-events --pull-request-id 8
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribePullRequestEvents](#) in AWS CLI Command Reference.

## **disassociate-approval-rule-template-from-repository**

Il seguente esempio di codice mostra come utilizzare `disassociate-approval-rule-template-from-repository`.

## AWS CLI

Per dissociare un modello di regola di approvazione da un repository

L'`disassociate-approval-rule-template-from-repository` seguente dissocia il modello di regola di approvazione specificato da un archivio denominato `MyDemoRepo`

```
aws codecommit disassociate-approval-rule-template-from-repository \  
  --repository-name MyDemoRepo \  
  --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Dissociazione di un modello di regola di approvazione](#) nella Guida per l'utente AWS CodeCommit

- Per API i dettagli, vedere [DisassociateApprovalRuleTemplateFromRepository](#) in AWS CLI Command Reference.

## `evaluate-pull-request-approval-rules`

Il seguente esempio di codice mostra come utilizzare `evaluate-pull-request-approval-rules`.

### AWS CLI

Per valutare se una pull request ha tutte le sue regole di approvazione soddisfatte

L'`evaluate-pull-request-approval-rules` seguente valuta lo stato delle regole di approvazione sulla pull request specificata. In questo esempio, una regola di approvazione non è stata soddisfatta per la pull request, quindi l'output del comando mostra un `approved` valore di `false`.

```
aws codecommit evaluate-pull-request-approval-rules \  
  --pull-request-id 27 \  
  --revision-id 9f29d167EXAMPLE
```

Output:

```
{  
  "evaluation": {  
    "approved": false,  
    "approvalRulesNotSatisfied": [  
      "Require two approved approvers"  
    ]  
  }  
}
```

```

    ],
    "overridden": false,
    "approvalRulesSatisfied": []
  }
}

```

Per ulteriori informazioni, consulta [Merge a Pull Request](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [EvaluatePullRequestApprovalRules](#) in AWS CLI Command Reference.

## get-approval-rule-template

Il seguente esempio di codice mostra come utilizzare `get-approval-rule-template`.

### AWS CLI

Per ottenere il contenuto di un modello di regola di approvazione

L'`get-approval-rule-template` esempio seguente ottiene il contenuto di un modello di regola di approvazione denominato `1-approver-rule-for-all-pull-requests`.

```

aws codecommit get-approval-rule-template \
  --approval-rule-template-name 1-approver-rule-for-all-pull-requests

```

Output:

```

{
  "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."
  }
}

```

Per ulteriori informazioni, vedere [Gestire i modelli di regole di approvazione](#) nella Guida AWS CodeCommit per l'utente.

- Per API i dettagli, vedere [GetApprovalRuleTemplate](#) in AWS CLI Command Reference.

## get-blob

Il seguente esempio di codice mostra come utilizzare `get-blob`.

### AWS CLI

Per visualizzare informazioni su un oggetto Git blob

L'`get-blob` esempio seguente recupera informazioni su un blob Git con l'ID '2eb4af3b EXAMPLE' in un repository denominato ' '. AWS CodeCommit MyDemoRepo

```
aws codecommit get-blob --repository-name MyDemoRepo --blob-id 2eb4af3bEXAMPLE
```

Output:

```
{
  "content": "QSBcaw5hcnkgTGFyToEXAMPLE="
}
```

- API Per i [GetBlob AWS CLI](#) dettagli, vedere in Command Reference.

## get-branch

Il seguente esempio di codice mostra come utilizzare `get-branch`.

### AWS CLI

Per ottenere informazioni su una filiale

Questo esempio ottiene informazioni su una filiale in un AWS CodeCommit repository.

Comando:

```
aws codecommit get-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```



**Output:**

```
{
  "BranchInfo": {
    "commitID": "317f8570EXAMPLE",
    "branchName": "MyNewBranch"
  }
}
```

- Per API i dettagli, vedere [GetBranchin](#) AWS CLI Command Reference.

**get-comment-reactions**

Il seguente esempio di codice mostra come utilizzare `get-comment-reactions`.

**AWS CLI**

Per visualizzare le reazioni emoji a un commento

L'`get-comment-reactions` esempio seguente elenca tutte le reazioni emoji a un commento con l'ID di `abcd1234EXAMPLEb5678efgh`. Se il carattere della shell supporta la visualizzazione della versione Emoji 1.0, nell'output `emoji` dell'emoji viene visualizzata.

```
aws codecommit get-comment-reactions \
  --comment-id abcd1234EXAMPLEb5678efgh
```

**Output:**

```
{
  "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"
    ]
  }
]
}
}

```

Per ulteriori informazioni, consulta [Commentare un commit AWS CodeCommit nella Guida](#) per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [GetCommentReactions](#) in AWS CLI Command Reference.

## get-comment

Il seguente esempio di codice mostra come utilizzare `get-comment`.

### AWS CLI

Per visualizzare i dettagli di un commento

Questo esempio dimostra come visualizzare i dettagli di un commento con l'ID del commento generato dal sistema di. `ff30b348EXAMPLEb9aa670f`

```
aws codecommit get-comment \
```

```
--comment-id ff30b348EXAMPLEb9aa670f
```

Output:

```
{
  "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
    }
  }
}
```

- Per API i dettagli, vedere [GetComment](#) in AWS CLI Command Reference.

## get-comments-for-compared-commit

Il seguente esempio di codice mostra come utilizzare `get-comments-for-compared-commit`.

### AWS CLI

Per visualizzare i commenti su un commit

Questo esempio dimostra come visualizzare i commenti fatti sul confronto tra due commit in un repository denominato `MyDemoRepo`

```
aws codecommit get-comments-for-compared-commit \
  --repository-name MyDemoRepo \
  --before-commit-ID 6e147360EXAMPLE \
  --after-commit-id 317f8570EXAMPLE
```

## Output:

```
{
  "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"
}

```

- Per API i dettagli, vedere [GetCommentsForComparedCommit](#) in AWS CLI Command Reference.

## get-comments-for-pull-request

Il seguente esempio di codice mostra come utilizzare `get-comments-for-pull-request`.

### AWS CLI

Per visualizzare i commenti relativi a una pull request

Questo esempio dimostra come visualizzare i commenti per una richiesta pull in un repository denominato `MyDemoRepo`

```

aws codecommit get-comments-for-pull-request \
  --repository-name MyDemoRepo \
  --before-commit-ID 317f8570EXAMPLE \
  --after-commit-id 5d036259EXAMPLE

```

Output:

```

{
  "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"
}

```

- Per API i dettagli, vedere [GetCommentsForPullRequest](#) in AWS CLI Command Reference.

## get-commit

Il seguente esempio di codice mostra come utilizzare `get-commit`.

### AWS CLI

Per visualizzare informazioni su un commit in un repository

Questo esempio mostra i dettagli su un commit con l'ID generato dal sistema '7e9fd3091thisisanexamplethisisanexample1' in un repository denominato ". AWS CodeCommit MyDemoRepo

Comando:

```
aws codecommit get-commit --repository-name MyDemoRepo --commit-id 7e9fd3091thisisanexamplethisisanexample1
```

Output:

```
{
  "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": "347a3408thisisanexampletreeidexample",
    "parents": [
      "7aa87a031thisisanexamplethisisanexample1"
    ],
    "message": "Fix incorrect variable name"
  }
}
```

- API Per i [GetCommit](#) dettagli AWS CLI , vedere in Command Reference.

## get-differences

Il seguente esempio di codice mostra come utilizzare `get-differences`.

### AWS CLI

Per ottenere informazioni sulle differenze relative a un identificatore di commit in un repository

Questo esempio mostra le informazioni sui metadati relative alle modifiche tra due specificatori di commit (branch, tag o altri riferimenti HEAD completi, come commitIDs) in una cartella rinominata nel repository denominata. AWS CodeCommit MyDemoRepo L'esempio include diverse opzioni non obbligatorie, tra cui `--before-commit-specifier`, `--before-path` e `--after-path`, per illustrare in modo più completo come utilizzare queste opzioni per limitare i risultati. La risposta include le autorizzazioni in modalità file.

Comando:

```
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
```

Output:

```
{
  "differences": [
    {
      "afterBlob": {
        "path": "blob.txt",
        "blobId": "2eb4af3b1thisisanexamplethisisanexample1",
        "mode": "100644"
      },
      "changeType": "M",
      "beforeBlob": {
        "path": "blob.txt",
        "blobId": "bf7fcf281thisisanexamplethisisanexample1",
        "mode": "100644"
      }
    }
  ]
}
```



- Per API i dettagli, vedere [GetDifferences](#) in AWS CLI Command Reference.

## get-file

Il seguente esempio di codice mostra come utilizzare `get-file`.

### AWS CLI

Per ottenere il contenuto codificato in base 64 di un file in un repository AWS CodeCommit

L'`get-file` esempio seguente mostra come ottenere il contenuto codificato in base 64 di un file denominato `README.md` da un ramo denominato `main` in un repository denominato `MyDemoRepo`

```
aws codecommit get-file \  
  --repository-name MyDemoRepo \  
  --commit-specifier main \  
  --file-path README.md
```

Output:

```
{  
  "blobId": "559b44fEXAMPLE",  
  "commitId": "c5709475EXAMPLE",  
  "fileContent": "IyBQaHVzEXAMPLE",  
  "filePath": "README.md",  
  "fileMode": "NORMAL",  
  "fileSize": 1563  
}
```

Per ulteriori informazioni, consultate la guida di riferimento [GetFile](#).AWS CodeCommit API

- Per API i dettagli, vedere [GetFile](#) in AWS CLI Command Reference.

## get-folder

Il seguente esempio di codice mostra come utilizzare `get-folder`.

### AWS CLI

Per ottenere il contenuto di una cartella in un AWS CodeCommit repository

L'get-folder esempio seguente mostra come ottenere il contenuto di una cartella di primo livello da un repository denominato. MyDemoRepo

```
aws codecommit get-folder --repository-name MyDemoRepo --folder-path ""
```

Output:

```
{
  "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"
}

```

Per ulteriori informazioni, consultate la guida [GetFolder](#) di riferimento. AWS CodeCommit API

- Per API i dettagli, vedere [GetFolder](#) in AWS CLI Command Reference.

## get-merge-commit

Il seguente esempio di codice mostra come utilizzare `get-merge-commit`.

### AWS CLI

Per ottenere informazioni dettagliate su un commit di unione

L'esempio seguente visualizza i dettagli su un commit di unione per il ramo di origine denominato `bugfix-bug1234` con un ramo di destinazione denominato `main` utilizzando la MERGE strategia `THREE_WAY_MERGE` in un repository denominato `MyDemoRepo`

```

aws codecommit get-merge-commit \
  --source-commit-specifier bugfix-bug1234 \
  --destination-commit-specifier main \
  --merge-option THREE_WAY_MERGE \
  --repository-name MyDemoRepo

```

Output:

```

{
  "sourceCommitId": "c5709475EXAMPLE",

```

```

    "destinationCommitId": "317f8570EXAMPLE",
    "baseCommitId": "fb12a539EXAMPLE",
    "mergeCommitId": "ffc4d608eEXAMPLE"
  }

```

Per ulteriori informazioni, vedere [View Commit Details nella Guida](#) per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [GetMergeCommit](#) in AWS CLI Command Reference.

## get-merge-conflicts

Il seguente esempio di codice mostra come utilizzare `get-merge-conflicts`.

### AWS CLI

Per vedere se ci sono conflitti di unione per una pull request

L'esempio seguente mostra se vi sono conflitti di fusione tra la punta di un ramo di origine denominato `feature-randomizationfeature` e un ramo di destinazione denominato «main» in un repository denominato `MyDemoRepo`

```

aws codecommit get-merge-conflicts \
  --repository-name MyDemoRepo \
  --source-commit-specifier feature-randomizationfeature \
  --destination-commit-specifier main \
  --merge-option THREE_WAY_MERGE

```

Output:

```

{
  "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"
    }
  }
]
}
```

- Per API i dettagli, vedere [GetMergeConflicts](#) in AWS CLI Command Reference.

## get-merge-options

Il seguente esempio di codice mostra come utilizzare `get-merge-options`.

### AWS CLI

Per ottenere informazioni sulle opzioni di unione disponibili per unire due rami specifici

L'`get-merge-options` esempio seguente determina le opzioni di unione disponibili per unire un ramo di origine denominato `bugfix-bug1234` con un ramo di destinazione denominato `main` in un repository denominato `MyDemoRepo`

```
aws codecommit get-merge-options \
```

```
--source-commit-specifier bugfix-bug1234 \  
--destination-commit-specifier main \  
--repository-name MyDemoRepo
```

Output:

```
{  
  "mergeOptions": [  
    "FAST_FORWARD_MERGE",  
    "SQUASH_MERGE",  
    "THREE_WAY_MERGE"  
  ],  
  "sourceCommitId": "18059494EXAMPLE",  
  "destinationCommitId": "ffd3311dEXAMPLE",  
  "baseCommitId": "ffd3311dEXAMPLE"  
}
```

Per ulteriori informazioni, consulta [Resolve Conflicts in a Pull Request nella Guida](#) per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [GetMergeOptions](#) in AWS CLI Command Reference.

## get-pull-request-approval-states

Il seguente esempio di codice mostra come utilizzare `get-pull-request-approval-states`.

AWS CLI

Per visualizzare le approvazioni su una pull request

L'`get-pull-request-approval-states` esempio seguente restituisce le approvazioni per la pull request specificata.

```
aws codecommit get-pull-request-approval-states \  
--pull-request-id 8 \  
--revision-id 9f29d167EXAMPLE
```

Output:

```
{  
  "approvals": [  
    {
```

```

        "userArn": "arn:aws:iam::123456789012:user/Mary_Major",
        "approvalState": "APPROVE"
    }
]
}

```

Per ulteriori informazioni, consulta [View Pull Requests](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [GetPullRequestApprovalStates](#) in AWS CLI Command Reference.

## get-pull-request-override-state

Il seguente esempio di codice mostra come utilizzare `get-pull-request-override-state`.

### AWS CLI

Per ottenere informazioni sullo stato di override di una pull request

L'`get-pull-request-override-state` esempio seguente restituisce lo stato di override per la pull request specificata. In questo esempio, le regole di approvazione per la pull request sono state sostituite da un utente di nome Mary Major, pertanto l'output restituisce un valore di `true` :

```

aws codecommit get-pull-request-override-state \
  --pull-request-id 34 \
  --revision-id 9f29d167EXAMPLE

```

Output:

```

{
  "overridden": true,
  "overrider": "arn:aws:iam::123456789012:user/Mary_Major"
}

```

Per ulteriori informazioni, consulta [Ignorare le regole di approvazione su una pull request](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [GetPullRequestOverrideState](#) in AWS CLI Command Reference.

## get-pull-request

Il seguente esempio di codice mostra come utilizzare `get-pull-request`.

## AWS CLI

Per visualizzare i dettagli di una pull request

Questo esempio dimostra come visualizzare le informazioni su una richiesta pull con l'ID di 27.

```
aws codecommit get-pull-request \  
  --pull-request-id 27
```

Output:

```
{  
  "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-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"
  }
}
```

- Per API i dettagli, vedere [GetPullRequest](#) in AWS CLI Command Reference.

## get-repository-triggers

Il seguente esempio di codice mostra come utilizzare `get-repository-triggers`.

### AWS CLI

Per ottenere informazioni sui trigger in un repository

Questo esempio mostra i dettagli sui trigger configurati per un AWS CodeCommit repository denominato `MyDemoRepo`.

```
aws codecommit get-repository-triggers \
  --repository-name MyDemoRepo
```

Output:

```
{
  "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"
        ]
    }
]
}

```

- Per API i dettagli, vedere [GetRepositoryTriggers](#) in AWS CLI Command Reference.

## get-repository

Il seguente esempio di codice mostra come utilizzare `get-repository`.

### AWS CLI

Per ottenere informazioni su un repository

Questo esempio mostra i dettagli su un AWS CodeCommit repository.

```

aws codecommit get-repository \
  --repository-name MyDemoRepo

```

Output:

```

{
  "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"  
  }  
}
```

- Per API i dettagli, vedere [GetRepository](#) in AWS CLI Command Reference.

## list-approval-rule-templates

Il seguente esempio di codice mostra come utilizzare `list-approval-rule-templates`.

### AWS CLI

Per elencare tutti i modelli di regole di approvazione in una AWS regione

L'`list-approval-rule-templates` esempio seguente elenca tutti i modelli di regole di approvazione nella regione specificata. Se non viene specificata alcuna AWS regione come parametro, il comando restituisce i modelli di regole di approvazione per l'area specificata nel AWS CLI profilo utilizzato per eseguire il comando.

```
aws codecommit list-approval-rule-templates \  
  --region us-east-2
```

Output:

```
{  
  "approvalRuleTemplateName": [  
    "2-approver-rule-for-main",  
    "1-approver-rule-for-all-pull-requests"  
  ]  
}
```

Per ulteriori informazioni, vedere [Gestire i modelli di regole di approvazione](#) nella Guida AWS CodeCommit per l'utente.

- Per API i dettagli, vedere [ListApprovalRuleTemplates](#) in AWS CLI Command Reference.

## list-associated-approval-rule-templates-for-repository

Il seguente esempio di codice mostra come utilizzare `list-associated-approval-rule-templates-for-repository`.

## AWS CLI

Per elencare tutti i modelli associati a un repository

L'`list-associated-approval-rule-templates-for-repository` seguente elenca tutti i modelli di regole di approvazione associati a un repository denominato `MyDemoRepo`.

```
aws codecommit list-associated-approval-rule-templates-for-repository \
  --repository-name MyDemoRepo
```

Output:

```
{
  "approvalRuleTemplateName": [
    "2-approver-rule-for-main",
    "1-approver-rule-for-all-pull-requests"
  ]
}
```

Per ulteriori informazioni, vedere [Gestire i modelli di regole di approvazione](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [ListAssociatedApprovalRuleTemplatesForRepository](#) in AWS CLI Command Reference.

## `list-branches`

Il seguente esempio di codice mostra come utilizzare `list-branches`.

## AWS CLI

Per visualizzare un elenco di nomi di filiali

Questo esempio elenca tutti i nomi delle filiali in un AWS CodeCommit repository.

```
aws codecommit list-branches \
  --repository-name MyDemoRepo
```

Output:

```
{
```

```
    "branches": [
      "MyNewBranch",
      "main"
    ]
  }
```

- Per API i dettagli, vedere [ListBranches](#) in AWS CLI Command Reference.

## list-pull-requests

Il seguente esempio di codice mostra come utilizzare `list-pull-requests`.

### AWS CLI

Per visualizzare un elenco di richieste pull in un repository

Questo esempio dimostra come elencare le richieste pull create da un IAM utente con ARN 'arn:aws:iam: :1111:user/LI\_JUAN' e lo stato di " in un repository denominato ": 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
```

Output:

```
{
  "nextToken": "",
  "pullRequestIds": ["2", "12", "16", "22", "23", "35", "30", "39", "47"]
}
```

- API [ListPullRequests](#) Per AWS CLI i dettagli, vedere in Command Reference.

## list-repositories-for-approval-rule-template

Il seguente esempio di codice mostra come utilizzare `list-repositories-for-approval-rule-template`.

### AWS CLI

Per elencare tutti i repository associati a un modello

L'`list-repositories-for-approval-rule-template` seguente elenca tutti i repository associati al modello di regola di approvazione specificato.

```
aws codecommit list-repositories-for-approval-rule-template \
  --approval-rule-template-name 2-approver-rule-for-main
```

Output:

```
{
  "repositoryNames": [
    "MyDemoRepo",
    "MyClonedRepo"
  ]
}
```

Per ulteriori informazioni, vedere [Gestire i modelli di regole di approvazione](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [ListRepositoriesForApprovalRuleTemplate](#) in AWS CLI Command Reference.

## **list-repositories**

Il seguente esempio di codice mostra come utilizzare `list-repositories`.

AWS CLI

Per visualizzare un elenco di repository

Questo esempio elenca tutti i AWS CodeCommit repository associati all'account dell' AWS utente.

Comando:

```
aws codecommit list-repositories
```

Output:

```
{
  "repositories": [
    {
      "repositoryName": "MyDemoRepo"
      "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",

```

```
    },
    {
      "repositoryName": "MyOtherDemoRepo"
      "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE"
    }
  ]
}
```

- Per API i dettagli, vedere [ListRepositories](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per visualizzare i AWS tag di un repository

L'`list-tags-for-resource` esempio seguente elenca le chiavi e i valori dei tag per il repository specificato.

```
aws codecommit list-tags-for-resource \
  --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo
```

Output:

```
{
  "tags": {
    "Status": "Secret",
    "Team": "Saanvi"
  }
}
```

Per ulteriori informazioni, vedete [View Tags for a Repository nella Guida](#) per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## merge-branches-by-fast-forward

Il seguente esempio di codice mostra come utilizzare `merge-branches-by-fast-forward`.

## AWS CLI

Per unire due rami utilizzando la strategia di fusione rapida

L'`merge-branches-by-fast-forward` esempio seguente unisce il ramo di origine specificato con il ramo di destinazione specificato in un repository denominato `MyDemoRepo`

```
aws codecommit merge-branches-by-fast-forward \  
  --source-commit-specifier bugfix-bug1234 \  
  --destination-commit-specifier bugfix-bug1233 \  
  --repository-name MyDemoRepo
```

Output:

```
{  
  "commitId": "4f178133EXAMPLE",  
  "treeId": "389765daEXAMPLE"  
}
```

Per ulteriori informazioni, consultate [Compare and Merge Branches](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [MergeBranchesByFastForward](#) in AWS CLI Command Reference.

## merge-branches-by-squash

Il seguente esempio di codice mostra come utilizzare `merge-branches-by-squash`.

## AWS CLI

Per unire due rami usando la strategia squash merge

L'`merge-branches-by-squash` esempio seguente unisce il ramo di origine specificato con il ramo di destinazione specificato in un repository denominato `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
```



Output:

```
{
  "commitId": "4f178133EXAMPLE",
  "treeId": "389765daEXAMPLE"
}
```

Per ulteriori informazioni, consultate [Compare and Merge Branches](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [MergeBranchesBySquash](#) in AWS CLI Command Reference.

### **merge-branches-by-three-way**

Il seguente esempio di codice mostra come utilizzare `merge-branches-by-three-way`.

AWS CLI

Per unire due rami utilizzando la strategia di fusione a tre vie

L'`merge-branches-by-three-way` esempio seguente unisce il ramo di origine specificato con il ramo di destinazione specificato in un repository denominato `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
```

Output:

```
{
  "commitId": "4f178133EXAMPLE",
  "treeId": "389765daEXAMPLE"
}
```

Per ulteriori informazioni, consultate [Compare and Merge Branches](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [MergeBranchesByThreeWay](#) in AWS CLI Command Reference.

## merge-pull-request-by-fast-forward

Il seguente esempio di codice mostra come utilizzare `merge-pull-request-by-fast-forward`.

### AWS CLI

Per unire e chiudere una pull request

Questo esempio dimostra come unire e chiudere una richiesta pull con l'ID «47» e un ID di commit di origine di «EXAMPLE99132ab0» in un repository denominato. `MyDemoRepo`

```
aws codecommit merge-pull-request-by-fast-forward \  
  --pull-request-id 47 \  
  --source-commit-id 99132ab0EXAMPLE \  
  --repository-name MyDemoRepo
```

Output:

```
{  
  "pullRequest": {  
    "approvalRules": [  
      {  
        "approvalRuleContent": "{\"Version\": \"2018-11-08\", \"Statements\":  
[[{\"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"
  }
}

```

Per ulteriori informazioni, consulta [Merge](#) a Pull Request nella Guida per l'utente AWS CodeCommit

- Per API i dettagli, vedere [MergePullRequestByFastForward](#) in AWS CLI Command Reference.

## merge-pull-request-by-squash

Il seguente esempio di codice mostra come utilizzare `merge-pull-request-by-squash`.

### AWS CLI

Per unire una richiesta pull usando la strategia squash merge

L'`merge-pull-request-by-squash` esempio seguente unisce e chiude la richiesta pull specificata utilizzando la strategia di risoluzione dei conflitti di `ACCEPT_SOURCE` in un repository denominato `MyDemoRepo`

```

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"
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Merge a Pull Request](#) nella Guida per l'utente.AWS CodeCommit

- Per API i dettagli, vedere [MergePullRequestBySquash](#) in AWS CLI Command Reference.

## merge-pull-request-by-three-way

Il seguente esempio di codice mostra come utilizzare `merge-pull-request-by-three-way`.

### AWS CLI

Per unire una richiesta pull utilizzando la strategia di unione a tre vie

L'`merge-pull-request-by-three-way` esempio seguente unisce e chiude la richiesta pull specificata utilizzando le opzioni predefinite per i dettagli dei conflitti e la strategia di risoluzione dei conflitti in un repository denominato. `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"

```

Output:

```

{
  "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"
}
}

```

Per ulteriori informazioni, consulta [Merge a Pull Request](#) nella Guida per l'utente.AWS CodeCommit

- Per API i dettagli, vedere [MergePullRequestByThreeWay](#) in AWS CLI Command Reference.

## override-pull-request-approval-rules

Il seguente esempio di codice mostra come utilizzare `override-pull-request-approval-rules`.

### AWS CLI

Per sovrascrivere i requisiti delle regole di approvazione in una pull request

L'`override-pull-request-approval-rules` esempio seguente sostituisce le regole di approvazione sulla pull request specificata. Per revocare invece un'override, impostate il valore del parametro su. `--override-status REVOKE`

```
aws codecommit override-pull-request-approval-rules \  
  --pull-request-id 34 \  
  --revision-id 927df8d8EXAMPLE \  
  --override-status OVERRIDE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Ignorare [le regole di approvazione su una pull request nella Guida](#) per l'utente.AWS CodeCommit

- Per API i dettagli, vedere [OverridePullRequestApprovalRules](#) in AWS CLI Command Reference.

## post-comment-for-compared-commit

Il seguente esempio di codice mostra come utilizzare `post-comment-for-compared-commit`.

### AWS CLI

Per creare un commento su un commit

Questo esempio dimostra come aggiungere il commento "Can you add a test case for this?" sulla modifica al `cl_sample.js` file nel confronto tra due commit in un repository denominato. MyDemoRepo

```
aws codecommit post-comment-for-compared-commit \  
  --repository-name MyDemoRepo \  
  --before-commit-id 317f8570EXAMPLE \  
  --comment "Can you add a test case for this?"
```

```
--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
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [PostCommentForComparedCommit](#) in AWS CLI Command Reference.

## post-comment-for-pull-request

Il seguente esempio di codice mostra come utilizzare `post-comment-for-pull-request`.

### AWS CLI

Per aggiungere un commento a una pull request



L'operazione `aws codecommit post-comment-for-pull-request` seguente aggiunge il commento «Questi non sembrano essere usati da nessuna parte. Possiamo rimuoverli?» sulla modifica del `ahs_count.py` file in una pull request con l'ID di 47 in un repository denominato `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
```

Output:

```
{
  "afterBlobId": "1f330709EXAMPLE",
  "afterCommitId": "5d036259EXAMPLE",
  "beforeBlobId": "80906a4cEXAMPLE",
  "beforeCommitId": "317f8570EXAMPLE",
  "comment": {
    "authorArn": "arn:aws:iam::111111111111:user/Saanvi_Sarkar",
    "clientRequestToken": "123Example",
    "commentId": "abcd1234EXAMPLEb5678efgh",
    "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"
}
```

- Per API i dettagli, vedere [PostCommentForPullRequest](#) in AWS CLI Command Reference.

## post-comment-reply

Il seguente esempio di codice mostra come utilizzare `post-comment-reply`.

### AWS CLI

Per rispondere a un commento su un commit o in una pull request

Questo esempio dimostra come aggiungere la risposta "Good catch. I'll remove them." al commento con l'ID generato dal sistema di. `abcd1234EXAMPLEb5678efgh`

```
aws codecommit post-comment-reply \  
  --in-reply-to abcd1234EXAMPLEb5678efgh \  
  --content "Good catch. I'll remove them." \  
  --client-request-token 123Example
```

Output:

```
{  
  "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": []  
  }  
}
```

- Per API i dettagli, vedere [PostCommentReply](#) in AWS CLI Command Reference.

## put-comment-reaction

Il seguente esempio di codice mostra come utilizzare `put-comment-reaction`.

## AWS CLI

Per rispondere a un commento su un commit con un'emoji

L'`put-comment-reaction` seguente risponde a un commento con l'ID di `abcd1234EXAMPLEb5678efgh` con un valore di reazione emoji pari a `:thumbsup:`:

```
aws codecommit put-comment-reaction \  
  --comment-id abcd1234EXAMPLEb5678efgh \  
  --reaction-value :thumbsup:
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Commentare un commit in AWS CodeCommit nella Guida](#) per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [PutCommentReaction](#) in AWS CLI Command Reference.

## put-file

Il seguente esempio di codice mostra come utilizzare `put-file`.

## AWS CLI

Per aggiungere un file a un repository

L'`put-file` seguente aggiunge un file denominato 'ExampleSolution.py' a un repository denominato " a un ramo denominato MyDemoRepo 'feature-randomizationfeature' il cui commit più recente ha un ID di '4c925148'. 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."
```

Output:

```
{
  "blobId": "2eb4af3bEXAMPLE",
  "commitId": "317f8570EXAMPLE",
  "treeId": "347a3408EXAMPLE"
}
```

- Per API i dettagli, vedere in Command Reference. [PutFile](#) AWS CLI

## put-repository-triggers

Il seguente esempio di codice mostra come utilizzare `put-repository-triggers`.

### AWS CLI

Per aggiungere o aggiornare un trigger in un repository

Questo esempio dimostra come aggiornare i trigger denominati " e 'MyFirstTriggerMySecondTrigger' utilizzando un JSON file già creato (qui denominato MyTriggers .json) che contiene la struttura di tutti i trigger per un repository denominato MyDemoRepo Per informazioni su come ottenere i trigger esistenti, consulta il comando. JSON `get-repository-triggers`

```
aws codecommit put-repository-triggers \
  --repository-name MyDemoRepo file://MyTriggers.json
```

Contenuto di 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"
    ]
  }
]
}

```

Output:

```

{
  "configurationId": "6fa51cd8-35c1-EXAMPLE"
}

```

- Per API i dettagli, vedere [PutRepositoryTriggers](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere AWS tag a un repository esistente

L'`tag-resource` esempio seguente contrassegna il repository specificato con due tag.

```

aws codecommit tag-resource \
  --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \
  --tags Status=Secret,Team=Saanvi

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere un tag a un repository nella Guida](#) per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## test-repository-triggers

Il seguente esempio di codice mostra come utilizzare `test-repository-triggers`.

### AWS CLI

Per testare i trigger in un repository

Questo esempio dimostra come testare un trigger denominato 'MyFirstTrigger' in un AWS CodeCommit repository denominato. MyDemoRepo In questo esempio, gli eventi nel repository attivano notifiche da un argomento di Amazon Simple Notification Service (AmazonSNS).

Comando:

```
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
```

Output:

```
{
  "successfulExecutions": [
    "MyFirstTrigger"
  ],
  "failedExecutions": []
}
```

- Per API i dettagli, consulta [TestRepositoryTriggers AWS CLI Command Reference](#).

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i AWS tag da un repository

L'`untag-resource` esempio seguente rimuove il tag con la chiave specificata dal repository denominato. MyDemoRepo

```
aws codecommit untag-resource \
--resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \
```

```
--tag-keys Status
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Rimuovere un tag da un repository nella Guida](#) per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-approval-rule-template-content

Il seguente esempio di codice mostra come utilizzare `update-approval-rule-template-content`.

### AWS CLI

Per aggiornare il contenuto di un modello di regola di approvazione

L'`update-approval-rule-template-content` esempio seguente modifica il contenuto del modello di regola di approvazione specificato per ridefinire il pool di approvazione per gli utenti che assumono il ruolo di `CodeCommitReview`.

```
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/*"}]}'
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, vedere [Gestire i modelli di regole di approvazione](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [UpdateApprovalRuleTemplateContent](#) in AWS CLI Command Reference.

## update-approval-rule-template-description

Il seguente esempio di codice mostra come utilizzare `update-approval-rule-template-description`.

### AWS CLI

Per aggiornare la descrizione di un modello di regola di approvazione

L'esempio seguente modifica la descrizione del modello di regola di approvazione specificato in `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"

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Gestire i modelli di regole di approvazione](#) nella Guida AWS CodeCommit per l'utente.

- Per API i dettagli, vedere [UpdateApprovalRuleTemplateDescription](#) in AWS CLI Command Reference.

## update-approval-rule-template-name

Il seguente esempio di codice mostra come utilizzare `update-approval-rule-template-name`.

### AWS CLI

Per aggiornare il nome di un modello di regola di approvazione

L'`update-approval-rule-template-name` esempio seguente modifica il nome di un modello di regola di approvazione da `1-approver-rule` a `1-approver-rule-for-all-pull-requests`.

```

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

```

Output:

```

{
  "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"
  }
}

```

```
}
```

Per ulteriori informazioni, vedere [Gestire i modelli di regole di approvazione](#) nella Guida per l'utente.AWS CodeCommit

- Per API i dettagli, vedere [UpdateApprovalRuleTemplateName](#)in AWS CLI Command Reference.

## update-comment

Il seguente esempio di codice mostra come utilizzare `update-comment`.

### AWS CLI

Per aggiornare un commento su un commit

Questo esempio dimostra come aggiungere il contenuto "Fixed as requested. I'll update the pull request." a un commento con un ID di `442b498bEXAMPLE5756813`.

```
aws codecommit update-comment \  
  --comment-id 442b498bEXAMPLE5756813 \  
  --content "Fixed as requested. I'll update the pull request."
```

Output:

```
{  
  "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  
      }  
  }  
}
```

- Per API i dettagli, vedere [UpdateComment](#) in AWS CLI Command Reference.

## update-default-branch

Il seguente esempio di codice mostra come utilizzare `update-default-branch`.

### AWS CLI

Per modificare il ramo predefinito per un repository

Questo esempio modifica il ramo predefinito per un AWS CodeCommit repository. Questo comando restituisce un output solo se ci sono errori.

Comando:

```
aws codecommit update-default-branch --repository-name MyDemoRepo --default-branch-name MyNewBranch
```

Output:

```
None.
```

- Per API i dettagli, vedere [UpdateDefaultBranch](#) in AWS CLI Command Reference.

## update-pull-request-approval-rule-content

Il seguente esempio di codice mostra come utilizzare `update-pull-request-approval-rule-content`.

### AWS CLI

Per modificare una regola di approvazione per una pull request

L'`update-pull-request-approval-rule-content` esempio seguente aggiorna la regola di approvazione specificata per richiedere l'approvazione di un utente da un pool di approvazioni che include qualsiasi IAM utente nell'123456789012 AWS account.

```
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/*\"]}]}"
```

Output:

```
{
  "approvalRule": {
    "approvalRuleContent": "{Version: 2018-11-08, Statements:
[\"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",
  }
}
```

Per ulteriori informazioni, consulta [Modificare o eliminare una regola di approvazione](#) nella Guida AWS CodeCommit per l'utente.

- Per API i dettagli, vedere [UpdatePullRequestApprovalRuleContent](#) in AWS CLI Command Reference.

## update-pull-request-approval-state

Il seguente esempio di codice mostra come utilizzare `update-pull-request-approval-state`.

AWS CLI

Per approvare o revocare l'approvazione di una pull request

L'`update-pull-request-approval-state` esempio seguente approva una pull request con l'ID di 27 e l'ID di revisione di 9f29d167EXAMPLE. Se invece desideri revocare l'approvazione, imposta il valore del `--approval-state` parametro su `REVOKE`.

```
aws codecommit update-pull-request-approval-state \
  --pull-request-id 27 \
  --revision-id 9f29d167EXAMPLE \
```

```
--approval-state "APPROVE"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Review a Pull Request](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [UpdatePullRequestApprovalState](#) in AWS CLI Command Reference.

## update-pull-request-description

Il seguente esempio di codice mostra come utilizzare `update-pull-request-description`.

### AWS CLI

Per modificare la descrizione di una pull request

Questo esempio dimostra come modificare la descrizione di una pull request con l'ID di 47.

```
aws codecommit update-pull-request-description \  
  --pull-request-id 47 \  
  --description "Updated the pull request to remove unused global variable."
```

Output:

```
{  
  "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"
  }
}

```

- Per API i dettagli, vedere [UpdatePullRequestDescription](#) in AWS CLI Command Reference.

## update-pull-request-status

Il seguente esempio di codice mostra come utilizzare `update-pull-request-status`.

### AWS CLI

Per modificare lo stato di una pull request

Questo esempio dimostra come modificare lo stato di una pull request con l'ID di in uno status 42 di CLOSED in un AWS CodeCommit repository denominato. MyDemoRepo

```

aws codecommit update-pull-request-status \
  --pull-request-id 42 \
  --pull-request-status CLOSED

```

Output:

```

{
  "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-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"
  }
}

```

- Per API i dettagli, vedere [UpdatePullRequestStatus](#) in AWS CLI Command Reference.

## update-pull-request-title

Il seguente esempio di codice mostra come utilizzare `update-pull-request-title`.

### AWS CLI

Per modificare il titolo di una pull request

Questo esempio dimostra come modificare il titolo di una pull request con l'ID di 47.

```

aws codecommit update-pull-request-title \
  --pull-request-id 47 \
  --title "Consolidation of global variables - updated review"

```

Output:

```

{
  "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"
  }
}

```

- Per API i dettagli, vedere [UpdatePullRequestTitle](#) in AWS CLI Command Reference.



## update-repository-description

Il seguente esempio di codice mostra come utilizzare `update-repository-description`.

### AWS CLI

Per modificare la descrizione di un repository

Questo esempio modifica la descrizione di un AWS CodeCommit repository. Questo comando restituisce un output solo se ci sono errori.

Comando:

```
aws codecommit update-repository-description --repository-name MyDemoRepo --  
repository-description "This description was changed"
```

Output:

```
None.
```

- Per API i dettagli, vedere [UpdateRepositoryDescription](#) in AWS CLI Command Reference.

## update-repository-name

Il seguente esempio di codice mostra come utilizzare `update-repository-name`.

### AWS CLI

Per modificare il nome di un repository

Questo esempio modifica il nome di un AWS CodeCommit repository. Questo comando restituisce un output solo se ci sono errori. La modifica del nome del AWS CodeCommit repository cambierà la SSH e HTTPS URLs che gli utenti devono connettersi al repository. Gli utenti non riusciranno a connettersi a questo repository finché non aggiorneranno le impostazioni di connessione. Inoltre, poiché il repository ARN cambierà, la modifica del nome del repository invaliderà tutte le politiche IAM utente che si basano su quel repository. ARN

Comando:

```
aws codecommit update-repository-name --old-name MyDemoRepo --new-  
name MyRenamedDemoRepo
```

## Output:

```
None .
```

- Per i API dettagli, consulta Command Reference. [UpdateRepositoryName](#) AWS CLI

## CodeDeploy esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with CodeDeploy.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **add-tags-to-on-premises-instances**

Il seguente esempio di codice mostra come utilizzare `add-tags-to-on-premises-instances`.

#### AWS CLI

Per aggiungere tag alle istanze locali

L'`add-tags-to-on-premises-instances` esempio seguente associa lo stesso tag AWS CodeDeploy di istanza locale a due istanze locali. Non registra le istanze locali con. AWS CodeDeploy

```
aws deploy add-tags-to-on-premises-instances \  
  --instance-names AssetTag12010298EX AssetTag23121309EX \  
  --tags Key=Name, Value=CodeDeployDemo-OnPrem
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [AddTagsToOnPremisesInstances](#) in AWS CLI Command Reference.

## batch-get-application-revisions

Il seguente esempio di codice mostra come utilizzare `batch-get-application-revisions`.

### AWS CLI

Per recuperare informazioni sulle revisioni delle applicazioni

L'`batch-get-application-revisions` seguente recupera le informazioni sulla revisione specificata memorizzate in un repository. GitHub

```
aws deploy batch-get-application-revisions \
  --application-name my-codedeploy-application \
  --revisions "[{"githubLocation": {"commitId":
  \"fa85936EXAMPLEa31736c051f10d77297EXAMPLE\", \"repository\": \"my-github-token/my-
  repository\"}, \"revisionType\": \"GitHub\"}]"
```

Output:

```
{
  "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": ""
}
```

Per ulteriori informazioni, vedere [BatchGetApplicationRevisions](#) nella sezione Reference.AWS CodeDeploy API

- Per API i dettagli, vedere [BatchGetApplicationRevisions](#) in AWS CLI Command Reference.

## batch-get-applications

Il seguente esempio di codice mostra come utilizzare `batch-get-applications`.

### AWS CLI

Per ottenere informazioni su più applicazioni

L'`batch-get-applications` esempio seguente visualizza informazioni su più applicazioni associate all' AWS account dell'utente.

```
aws deploy batch-get-applications --application-names WordPress_App MyOther_App
```

Output:

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [BatchGetApplications](#) in AWS CLI Command Reference.

## batch-get-deployment-groups

Il seguente esempio di codice mostra come utilizzare `batch-get-deployment-groups`.

### AWS CLI

Per recuperare informazioni su uno o più gruppi di distribuzione

L'esempio seguente recupera informazioni su due gruppi di distribuzione associati all'applicazione specificata CodeDeploy .

```
aws deploy batch-get-deployment-groups \
  --application-name my-codedeploy-application \
  --deployment-group-names ["my-deployment-group-1","my-deployment-group-2"]
```

Output:

```
{
  "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": ""
  }

```

Per ulteriori informazioni, vedere [BatchGetDeploymentGroups](#) nella Guida di AWS CodeDeploy API riferimento.

- Per API i dettagli, vedere [BatchGetDeploymentGroups](#) in AWS CLI Command Reference.

## batch-get-deployment-targets

Il seguente esempio di codice mostra come utilizzare `batch-get-deployment-targets`.

### AWS CLI

Per recuperare le destinazioni associate a una distribuzione

L'`batch-get-deployment-targets` esempio seguente restituisce informazioni su una delle destinazioni associate alla distribuzione specificata.

```

aws deploy batch-get-deployment-targets \
  --deployment-id "d-1A2B3C4D5" \
  --target-ids "i-01a2b3c4d5e6f1111"

```

Output:

```

{
  "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"
}
]
}

```

Per ulteriori informazioni, vedere [BatchGetDeploymentTargets](#) nella sezione AWS CodeDeploy API Reference.

- Per API i dettagli, vedere [BatchGetDeploymentTargets](#) in AWS CLI Command Reference.



## batch-get-deployments

Il seguente esempio di codice mostra come utilizzare `batch-get-deployments`.

### AWS CLI

Per ottenere informazioni su più distribuzioni

L'esempio seguente visualizza informazioni su più distribuzioni associate all'account dell'utente. AWS

```
aws deploy batch-get-deployments --deployment-ids d-A1B2C3111 d-A1B2C3222
```

Output:

```
{
  "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
    }
  ]
}

```

- Per API i dettagli, vedere [BatchGetDeployments](#) in AWS CLI Command Reference.

## batch-get-on-premises-instances

Il seguente esempio di codice mostra come utilizzare `batch-get-on-premises-instances`.

### AWS CLI

Per ottenere informazioni su una o più istanze locali

L'batch-get-on-premises-instances seguente ottiene informazioni su due istanze locali.

```
aws deploy batch-get-on-premises-instances --instance-  
names AssetTag12010298EX AssetTag23121309EX
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [BatchGetOnPremisesInstances](#) in AWS CLI Command Reference.

## continue-deployment

Il seguente esempio di codice mostra come utilizzare `continue-deployment`.

### AWS CLI

Per iniziare a reindirizzare il traffico senza attendere che scada il tempo di attesa specificato.

L'esempio seguente inizia a reindirizzare il traffico dalle istanze dell'ambiente originale che sono pronte per iniziare a spostare il traffico verso le istanze nell'ambiente sostitutivo.

```
aws deploy continue-deployment \  
  --deployment-id "d-A1B2C3111" \  
  --deployment-wait-type "READY_WAIT"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere nella Guida di riferimento. [ContinueDeployment](#) AWS CodeDeploy API

- Per API i dettagli, vedere [ContinueDeployment](#) in AWS CLI Command Reference.

## create-application

Il seguente esempio di codice mostra come utilizzare `create-application`.

### AWS CLI

Per creare un'applicazione

L'esempio seguente crea un'applicazione e la associa all'AWS account dell'utente.

```
aws deploy create-application --application-name MyOther_App
```

Output:

```
{  
  "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"  
}
```

- Per API i dettagli, vedere [CreateApplication](#) in AWS CLI Command Reference.

## create-deployment-config

Il seguente esempio di codice mostra come utilizzare `create-deployment-config`.

### AWS CLI

Per creare una configurazione di distribuzione personalizzata

L'`create-deployment-config` esempio seguente crea una configurazione di distribuzione personalizzata e la associa all' AWS account dell'utente.

```
aws deploy create-deployment-config \  
  --deployment-config-name ThreeQuartersHealthy \  
  --minimum-healthy-hosts type=FLEET_PERCENT,value=75
```

Output:

```
{  
  "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"  
}
```

- Per API i dettagli, vedere [CreateDeploymentConfig](#) in AWS CLI Command Reference.

## create-deployment-group

Il seguente esempio di codice mostra come utilizzare `create-deployment-group`.

### AWS CLI

Per creare un gruppo di distribuzione

L'`create-deployment-group` esempio seguente crea un gruppo di distribuzione e lo associa all'applicazione specificata e all' AWS account dell'utente.

```
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
```

Output:

```
{
  "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- Per API i dettagli, vedere [CreateDeploymentGroup](#) in AWS CLI Command Reference.

## create-deployment

Il seguente esempio di codice mostra come utilizzare `create-deployment`.

### AWS CLI

Esempio 1: creare una CodeDeploy distribuzione utilizzando la piattaforma di calcolo EC2 /On-premises

L'`create-deployment` esempio seguente crea una distribuzione e la associa all'account dell'utente. AWS

```
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
```

Output:

```
{
  "deploymentId": "d-A1B2C3111"
}
```

Esempio 2: creare una CodeDeploy distribuzione utilizzando la piattaforma di ECS calcolo Amazon

L'`create-deployment` esempio seguente utilizza i due file seguenti per distribuire un ECS servizio Amazon.

## Contenuto del `create-deployment.json` file:

```
{
  "applicationName": "ecs-deployment",
  "deploymentGroupName": "ecs-deployment-dg",
  "revision": {
    "revisionType": "S3",
    "s3Location": {
      "bucket": "ecs-deployment-bucket",
      "key": "appspec.yaml",
      "bundleType": "YAML"
    }
  }
}
```

Quel file, a sua volta, recupera il seguente file `appspec.yaml` da un bucket S3 chiamato `ecs-deployment-bucket`

```
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"
```

## Comando:

```
aws deploy create-deployment \
  --cli-input-json file://create-deployment.json \
  --region us-east-1
```

## Output:

```
{
  "deploymentId": "d-1234ABCDE"
}
```

Per ulteriori informazioni, consulta la sezione Reference [CreateDeployment](#).AWS CodeDeploy API

- Per API i dettagli, vedere [CreateDeployment](#) in AWS CLI Command Reference.

## delete-application

Il seguente esempio di codice mostra come utilizzare `delete-application`.

### AWS CLI

Eliminazione di un'applicazione

L'`delete-application` esempio seguente elimina l'applicazione specificata associata all' AWS account dell'utente.

```
aws deploy delete-application --application-name WordPress_App
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteApplication](#) in AWS CLI Command Reference.

## delete-deployment-config

Il seguente esempio di codice mostra come utilizzare `delete-deployment-config`.

### AWS CLI

Per eliminare una configurazione di distribuzione

L'`delete-deployment-config` esempio seguente elimina una configurazione di distribuzione personalizzata associata all' AWS account dell'utente.

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteDeploymentConfig](#) in AWS CLI Command Reference.

## delete-deployment-group

Il seguente esempio di codice mostra come utilizzare `delete-deployment-group`.



## AWS CLI

Per eliminare un gruppo di distribuzione

L'`delete-deployment-group` seguente elimina un gruppo di distribuzione associato all'applicazione specificata.

```
aws deploy delete-deployment-group \  
  --application-name WordPress_App \  
  --deployment-group-name WordPress_DG
```

Output:

```
{  
  "hooksNotCleanedUp": []  
}
```

- Per API i dettagli, vedere [DeleteDeploymentGroup](#) in AWS CLI Command Reference.

## `delete-git-hub-account-token`

Il seguente esempio di codice mostra come utilizzare `delete-git-hub-account-token`.

## AWS CLI

To elimina la connessione di un GitHub account

L'`delete-git-hub-account-token` seguente elimina la connessione dell'account specificato GitHub .

```
aws deploy delete-git-hub-account-token --token-name my-github-account
```

Output:

```
{  
  "tokenName": "my-github-account"  
}
```

Per ulteriori informazioni, vedere [DeleteGitHubAccountToken](#) nel AWS CodeDeploy API Reference.

- Per API i dettagli, vedere [DeleteGitHubAccountToken](#) in AWS CLI Command Reference.

## deregister-on-premises-instance

Il seguente esempio di codice mostra come utilizzare `deregister-on-premises-instance`.

### AWS CLI

Per annullare la registrazione di un'istanza locale

L'`deregister-on-premises-instance` seguente esempio annulla la registrazione di un'istanza locale con AWS CodeDeploy, ma non elimina l'IAM utente associato all'istanza, né si dissocia nei AWS CodeDeploy tag dell'istanza locale dall'istanza. Inoltre, non disinstalla l' AWS CodeDeploy agente dall'istanza né rimuove il file di configurazione locale dall'istanza.

```
aws deploy deregister-on-premises-instance --instance-name AssetTag12010298EX
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeregisterOnPremisesInstance](#) in AWS CLI Command Reference.

## deregister

Il seguente esempio di codice mostra come utilizzare `deregister`.

### AWS CLI

Per annullare la registrazione di un'istanza locale

L'`deregister` seguente esempio annulla la registrazione di un'istanza locale con. AWS CodeDeploy Non elimina l'IAM utente associato all'istanza. Nei tag locali si dissocia AWS CodeDeploy dall'istanza. Non disinstalla l' AWS CodeDeploy agente dall'istanza né rimuove il file di configurazione locale dall'istanza.

```
aws deploy deregister \  
  --instance-name AssetTag12010298EX \  
  --no-delete-iam-user \  
  --region us-west-2
```

Output:

```
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
```

- Per API i dettagli, consulta [Annullare la registrazione in Command ReferenceAWS CLI](#).

## get-application-revision

Il seguente esempio di codice mostra come utilizzare `get-application-revision`

### AWS CLI

Per ottenere informazioni sulla revisione di un'applicazione

L'`get-application-revision` seguente visualizza informazioni sulla revisione di un'applicazione associata all'applicazione specificata.

```
aws deploy get-application-revision \
  --application-name WordPress_App \
  --s3-
location bucket=CodeDeployDemoBucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,ke
```

Output:

```
{
  "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"
    }
}
```

- Per API i dettagli, vedere [GetApplicationRevision](#) in AWS CLI Command Reference.

## get-application

Il seguente esempio di codice mostra come utilizzare `get-application`.

### AWS CLI

Per ottenere informazioni su un'applicazione

L'`get-application` esempio seguente visualizza informazioni su un'applicazione associata all'AWS account dell'utente.

```
aws deploy get-application --application-name WordPress_App
```

Output:

```
{
  "application": {
    "applicationName": "WordPress_App",
    "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "createTime": 1407878168.078,
    "linkedToGitHub": false
  }
}
```

- Per API i dettagli, vedere [GetApplication](#) in AWS CLI Command Reference.

## get-deployment-config

Il seguente esempio di codice mostra come utilizzare `get-deployment-config`.

### AWS CLI

Per ottenere informazioni su una configurazione di distribuzione

L'`get-deployment-config` seguente visualizza informazioni su una configurazione di distribuzione associata all' AWS account dell'utente.

```
aws deploy get-deployment-config --deployment-config-name ThreeQuartersHealthy
```

Output:

```
{
  "deploymentConfigInfo": {
    "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "minimumHealthyHosts": {
      "type": "FLEET_PERCENT",
      "value": 75
    },
    "createTime": 1411081164.379,
    "deploymentConfigName": "ThreeQuartersHealthy"
  }
}
```

- Per API i dettagli, vedere [GetDeploymentConfig](#) in AWS CLI Command Reference.

## get-deployment-group

Il seguente esempio di codice mostra come utilizzare `get-deployment-group`.

### AWS CLI

Per visualizzare informazioni su un gruppo di distribuzione

L'`get-deployment-group` seguente visualizza informazioni su un gruppo di distribuzione associato all'applicazione specificata.

```
aws deploy get-deployment-group \
  --application-name WordPress_App \
  --deployment-group-name WordPress_DG
```

Output:

```
{
  "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"
}
}

```

- Per API i dettagli, vedere [GetDeploymentGroup](#) in AWS CLI Command Reference.

## get-deployment-instance

Il seguente esempio di codice mostra come utilizzare `get-deployment-instance`.

### AWS CLI

Per ottenere informazioni su un'istanza di distribuzione

L'`get-deployment-instance` esempio seguente visualizza informazioni su un'istanza di distribuzione associata alla distribuzione specificata.

```
aws deploy get-deployment-instance --deployment-id d-QA4G4F9EX --instance-id i-902e9fEX
```

Output:

```

{
  "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"
}
}
```

- Per API i dettagli, vedere [GetDeploymentInstance](#) in AWS CLI Command Reference.

## get-deployment-target

Il seguente esempio di codice mostra come utilizzare `get-deployment-target`.

### AWS CLI

Per restituire informazioni su un obiettivo di distribuzione

L'esempio seguente restituisce informazioni su un obiettivo di distribuzione associato alla distribuzione specificata.

```
aws deploy get-deployment-target \
  --deployment-id "d-A1B2C3111" \
  --target-id "i-a1b2c3d4e5f611111"
```

Output:

```
{
  "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"
}
}
```

Per ulteriori informazioni, vedere [GetDeploymentTarget](#) nella Guida di AWS CodeDeploy API riferimento.

- Per API i dettagli, vedere [GetDeploymentTarget](#) in AWS CLI Command Reference.

## get-deployment

Il seguente esempio di codice mostra come utilizzare `get-deployment`.

### AWS CLI

Per ottenere informazioni su una distribuzione

L'get-deploymentesempio seguente visualizza informazioni su una distribuzione associata all'AWS account dell'utente.

```
aws deploy get-deployment --deployment-id d-A1B2C3123
```

Output:

```
{
  "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
  }
}
```

- Per API i dettagli, vedere [GetDeployment](#) in AWS CLI Command Reference.

## get-on-premises-instance

Il seguente esempio di codice mostra come utilizzare `get-on-premises-instance`.

### AWS CLI

Per ottenere informazioni su un'istanza locale

L'esempio seguente recupera informazioni sull'istanza locale specificata.

```
aws deploy get-on-premises-instance --instance-name AssetTag12010298EX
```

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [GetOnPremisesInstance](#) in AWS CLI Command Reference.

## install

Il seguente esempio di codice mostra come utilizzare `install`.

### AWS CLI

Per installare un'istanza locale

L'installazione seguente copia il file di configurazione locale dalla posizione specificata sull'istanza alla posizione sull'istanza in cui l' AWS CodeDeploy agente si aspetta di trovarlo. Inoltre, installa l' AWS CodeDeploy agente sull'istanza. Non crea alcun IAM utente, né registra l'istanza locale né associa alcun tag di istanza locale all' AWS CodeDeploy istanza. AWS CodeDeploy

```
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
```

Output:

```
Creating the on-premises instance configuration file... DONE  
Installing the AWS CodeDeploy Agent... DONE
```

- Per API i dettagli, consulta [Install](#) in AWS CLI Command Reference.

## list-application-revisions

Il seguente esempio di codice mostra come utilizzare `list-application-revisions`.

AWS CLI

Per ottenere informazioni sulle revisioni delle applicazioni

L'`list-application-revisions` seguente visualizza informazioni su tutte le revisioni dell'applicazione associate all'applicazione specificata.

```
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
```

Output:

```
{
```

```
"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"  
    }  
  }  
]  
}
```

- Per API i dettagli, vedere [ListApplicationRevisions](#) in AWS CLI Command Reference.

## list-applications

Il seguente esempio di codice mostra come utilizzare `list-applications`.

### AWS CLI

Per ottenere informazioni sulle applicazioni

L'`list-applications` esempio seguente visualizza informazioni su tutte le applicazioni associate all' AWS account dell'utente.

```
aws deploy list-applications
```

Output:

```
{  
  "applications": [  
    "WordPress_App",  
  ]  
}
```

```
    "MyOther_App"  
  ]  
}
```

- Per API i dettagli, vedere [ListApplications](#) in AWS CLI Command Reference.

## list-deployment-configs

Il seguente esempio di codice mostra come utilizzare `list-deployment-configs`.

### AWS CLI

Per ottenere informazioni sulle configurazioni di distribuzione

L'`list-deployment-configs` esempio seguente visualizza informazioni su tutte le configurazioni di distribuzione associate all'account dell' AWS utente.

```
aws deploy list-deployment-configs
```

Output:

```
{  
  "deploymentConfigsList": [  
    "ThreeQuartersHealthy",  
    "CodeDeployDefault.AllAtOnce",  
    "CodeDeployDefault.HalfAtATime",  
    "CodeDeployDefault.OneAtATime"  
  ]  
}
```

- Per API i dettagli, vedere [ListDeploymentConfigs](#) in AWS CLI Command Reference.

## list-deployment-groups

Il seguente esempio di codice mostra come utilizzare `list-deployment-groups`.

### AWS CLI

Per ottenere informazioni sui gruppi di distribuzione

L'`list-deployment-groups` esempio seguente visualizza informazioni su tutti i gruppi di distribuzione associati all'applicazione specificata.

```
aws deploy list-deployment-groups --application-name WordPress_App
```

Output:

```
{
  "applicationName": "WordPress_App",
  "deploymentGroups": [
    "WordPress_DG",
    "WordPress_Beta_DG"
  ]
}
```

- Per API i dettagli, vedere [ListDeploymentGroups](#) in AWS CLI Command Reference.

## list-deployment-instances

Il seguente esempio di codice mostra come utilizzare `list-deployment-instances`.

AWS CLI

Per ottenere informazioni sulle istanze di distribuzione

L'`list-deployment-instances` esempio seguente visualizza informazioni su tutte le istanze di distribuzione associate alla distribuzione specificata.

```
aws deploy list-deployment-instances \
  --deployment-id d-A1B2C3111 \
  --instance-status-filter Succeeded
```

Output:

```
{
  "instancesList": [
    "i-EXAMPLE11",
    "i-EXAMPLE22"
  ]
}
```

- Per API i dettagli, vedere [ListDeploymentInstances](#) in AWS CLI Command Reference.



## list-deployment-targets

Il seguente esempio di codice mostra come utilizzare `list-deployment-targets`.

### AWS CLI

Per recuperare un elenco di destinazioni IDs associate a una distribuzione

L'`list-deployment-targets` esempio seguente recupera un elenco di oggetti IDs associati a distribuzioni con lo stato «Failed» o «InProgress».

```
aws deploy list-deployment-targets \
  --deployment-id "d-A1B2C3111" \
  --target-filters "{\"TargetStatus\":[\"Failed\",\"InProgress\"]}"
```

Output:

```
{
  "targetIds": [
    "i-0f1558aaf90e5f1f9"
  ]
}
```

Per ulteriori informazioni, vedere [ListDeploymentTargets](#) nella Guida di riferimento.AWS CodeDeploy API

- Per API i dettagli, vedere [ListDeploymentTargets](#) in AWS CLI Command Reference.

## list-deployments

Il seguente esempio di codice mostra come utilizzare `list-deployments`.

### AWS CLI

Per ottenere informazioni sulle distribuzioni

L'`list-deployments` esempio seguente visualizza informazioni su tutte le distribuzioni associate all'applicazione e al gruppo di distribuzione specificati.

```
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
```

Output:

```
{  
  "deployments": [  
    "d-EXAMPLE11",  
    "d-EXAMPLE22",  
    "d-EXAMPLE33"  
  ]  
}
```

- Per API i dettagli, vedere [ListDeployments](#) in AWS CLI Command Reference.

## list-git-hub-account-token-names

Il seguente esempio di codice mostra come utilizzare `list-git-hub-account-token-names`.

AWS CLI

Elenca i nomi delle connessioni memorizzate agli GitHub account

L'`list-git-hub-account-token-names` esempio seguente elenca i nomi delle connessioni memorizzate agli GitHub account AWS dell'utente corrente.

```
aws deploy list-git-hub-account-token-names
```

Output:

```
{  
  "tokenNameList": [  
    "my-first-token",  
    "my-second-token",  
    "my-third-token"  
  ]  
}
```

Per ulteriori informazioni, vedere [ListGitHubAccountTokenNames](#) nel AWS CodeDeploy API Reference.

- Per API i dettagli, vedere [ListGitHubAccountTokenNames](#) in AWS CLI Command Reference.

## list-on-premises-instances

Il seguente esempio di codice mostra come utilizzare `list-on-premises-instances`.

### AWS CLI

Per ottenere informazioni su una o più istanze locali

L'`list-on-premises-instances`esempio seguente recupera un elenco di nomi di istanze locali disponibili per le istanze registrate in AWS CodeDeploy e a cui è associato anche il tag di istanza locale specificato. AWS CodeDeploy

```
aws deploy list-on-premises-instances \
  --registration-status Registered \
  --tag-filters Key=Name,Value=CodeDeployDemo-OnPrem,Type=KEY_AND_VALUE
```

Output:

```
{
  "instanceNames": [
    "AssetTag12010298EX"
  ]
}
```

- Per i API dettagli, vedere [ListOnPremisesInstances](#)in Command Reference.AWS CLI

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una risorsa (applicazione)

L'`list-tags-for-resource`esempio seguente elenca i tag applicati a un'applicazione denominata `testApp` in CodeDeploy.

```
aws deploy list-tags-for-resource \
  --resource-arn arn:aws:codedeploy:us-west-2:111122223333:application:testApp
```

Output:

```
{
  "Tags": [
    {
      "Key": "Type",
      "Value": "testType"
    },
    {
      "Key": "Name",
      "Value": "testName"
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione [Tagging delle istanze per i gruppi di distribuzione CodeDeploy nella Guida](#) per l'AWS CodeDeploy utente.

- Per API i dettagli, consulta AWS CLI Command [ListTagsForResource](#)Reference.

## push

Il seguente esempio di codice mostra come utilizzare `push`.

### AWS CLI

Per raggruppare e distribuire una revisione dell'applicazione AWS CodeDeploy compatibile su Amazon S3

L'pushesempio seguente raggruppa e distribuisce una revisione dell'applicazione in Amazon S3 e quindi associa la revisione dell'applicazione all'applicazione specificata.

```
aws deploy push \
  --application-name WordPress_App \
  --description "This is my deployment" \
  --ignore-hidden-files \
  --s3-location s3://CodeDeployDemoBucket/WordPressApp.zip \
  --source /tmp/MyLocalDeploymentFolder/
```

L'output descrive come utilizzare il `create-deployment` comando per creare una distribuzione che utilizza la revisione dell'applicazione caricata.

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"
```

- Per API i dettagli, consulta [Push](#) in AWS CLI Command Reference.

## register-application-revision

Il seguente esempio di codice mostra come utilizzare `register-application-revision`.

### AWS CLI

Per registrare informazioni su una revisione di un'applicazione già caricata

L'esempio seguente registra le informazioni su una revisione di un'applicazione già caricata archiviata in Amazon S3 con 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
```

Questo comando non produce alcun output.

- Per i API dettagli, consulta Command Reference. [RegisterApplicationRevision](#) AWS CLI

## register-on-premises-instance

Il seguente esempio di codice mostra come utilizzare `register-on-premises-instance`.

### AWS CLI

Per registrare un'istanza locale

L'esempio seguente registra un'istanza locale con AWS CodeDeploy. Non crea l'IAM utente specificato né associa in AWS CodeDeploy nessuna istanza locale i tag all'istanza registrata.

```
aws deploy register-on-premises-instance \
```

```
--instance-name AssetTag12010298EX \  
--iam-user-arn arn:aws:iam::80398EXAMPLE:user/CodeDeployDemoUser-OnPrem
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [RegisterOnPremisesInstance](#) in AWS CLI Command Reference.

## register

Il seguente esempio di codice mostra come utilizzare `register`.

### AWS CLI

Per registrare un'istanza locale

L'esempio seguente registra un'istanza locale con AWS CodeDeploy, la associa AWS CodeDeploy all'istanza registrata nel tag di istanza locale specificato e crea un file di configurazione locale che può essere copiato nell'istanza. Non crea l'IAM utente né installa l'agente sull'istanza. AWS CodeDeploy

```
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
```

Output:

```
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
```

- Per API i dettagli, consulta [Register](#) in AWS CLI Command Reference.

## remove-tags-from-on-premises-instances

Il seguente esempio di codice mostra come utilizzare `remove-tags-from-on-premises-instances`.

## AWS CLI

Per rimuovere i tag da una o più istanze locali

L'`remove-tags-from-on-premises-instances` esempio seguente dissocia i tag locali specificati dalle istanze locali. AWS CodeDeploy Non annulla la registrazione delle istanze locali né disinstalla l' AWS CodeDeploy agente dall'istanza AWS CodeDeploy, né rimuove il file di configurazione locale dalle istanze, né elimina gli utenti associati alle istanze. IAM

```
aws deploy remove-tags-from-on-premises-instances \  
  --instance-names AssetTag12010298EX AssetTag23121309EX \  
  --tags Key=Name, Value=CodeDeployDemo-OnPrem
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere in Command Reference. [RemoveTagsFromOnPremisesInstances](#) AWS CLI

## stop-deployment

Il seguente esempio di codice mostra come utilizzare `stop-deployment`.

### AWS CLI

Per tentare di interrompere una distribuzione

L'`stop-deployment` esempio seguente tenta di interrompere una distribuzione in corso associata all' AWS account dell'utente.

```
aws deploy stop-deployment --deployment-id d-A1b2c3111
```

Output:

```
{  
  "status": "Succeeded",  
  "statusMessage": "No more commands will be scheduled for execution in the  
  deployment instances"  
}
```

- Per API i dettagli, vedi in Command [StopDeployment](#) Reference AWS CLI .

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Etichettare una risorsa (applicazione)

L'`tag-resource` esempio seguente aggiunge due tag con le chiavi `Name` e `Type` e valori `testName` e `testType` a un'applicazione denominata `testApp` in `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
```

In caso di successo, questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging delle istanze per i gruppi di distribuzione CodeDeploy nella Guida](#) per l'AWS CodeDeploy utente.

- Per API i dettagli, consulta AWS CLI Command [TagResource](#) Reference.

## uninstall

Il seguente esempio di codice mostra come utilizzare `uninstall`.

### AWS CLI

Per disinstallare un'istanza locale

L'`uninstall` esempio seguente disinstalla l' AWS CodeDeploy agente dall'istanza locale e rimuove il file di configurazione locale dall'istanza. Non annulla la registrazione dell'istanza AWS CodeDeploy, né dissocia i tag dell'istanza locale AWS CodeDeploy dall'istanza, né elimina l'utente associato all'IAM istanza.

```
aws deploy uninstall
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [Disinstalla](#) in Command Reference.AWS CLI



## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una risorsa (applicazione)

L'esempio seguente rimuove due tag con le chiavi `Name` e `Type` da un'applicazione denominata `testApp` in CodeDeploy.

```
aws deploy untag-resource \  
  --resource-arn arn:aws:codedeploy:us-west-2:111122223333:application:testApp \  
  --tag-keys Name Type
```

In caso di successo, questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging delle istanze per i gruppi di distribuzione CodeDeploy nella Guida](#) per l'AWS CodeDeploy utente.

- Per API i dettagli, consulta AWS CLI Command [UntagResource](#) Reference.

## update-application

Il seguente esempio di codice mostra come utilizzare `update-application`.

### AWS CLI

Per modificare i dettagli di un'applicazione

L'esempio seguente modifica il nome di un'applicazione associata all'AWS account dell'utente.

```
aws deploy update-application \  
  --application-name WordPress_App \  
  --new-application-name My_WordPress_App
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [UpdateApplication](#) in AWS CLI Command Reference.

## update-deployment-group

Il seguente esempio di codice mostra come utilizzare `update-deployment-group`.

### AWS CLI

Per modificare le informazioni su un gruppo di distribuzione

L'update-deployment-groupesempio seguente modifica le impostazioni di un gruppo di distribuzione associato all'applicazione specificata.

```
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
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [UpdateDeploymentGroup](#) in AWS CLI Command Reference.

## CodeGuru Esempi di revisori che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with CodeGuru Reviewer.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **associate-repository**

Il seguente esempio di codice mostra come utilizzare `associate-repository`.

#### AWS CLI

Esempio 1: per creare un'associazione di repository Bitbucket

L'esempio seguente crea un'associazione di repository utilizzando un repository Bitbucket esistente.

```
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 }'
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Creare un'associazione di repository Bitbucket in Amazon CodeGuru Reviewer nella Amazon CodeGuru Reviewer User Guide](#). CodeGuru

Esempio 2: creare un'associazione di repository aziendali GitHub

L'associate-repositoryesempio seguente crea un'associazione di repository utilizzando un repository GitHub Enterprise esistente.

```
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 }'
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Creare un'associazione di repository GitHub Enterprise Server in Amazon CodeGuru Reviewer nella Amazon CodeGuru Reviewer User Guide](#).

Esempio 3: creare un'associazione di repository AWS CodeCommit

L'associate-repositoryesempio seguente crea un'associazione di repository utilizzando un repository esistente AWS CodeCommit .

```
aws codeguru-reviewer associate-repository \
  --repository CodeCommit={Name=mySampleRepo}
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Creare un'associazione di AWS CodeCommit repository in Amazon CodeGuru Reviewer nella Amazon CodeGuru Reviewer User Guide](#).

- Per API i dettagli, consulta Command [AssociateRepository](#) Reference AWS CLI .

## create-code-review

Il seguente esempio di codice mostra come utilizzare `create-code-review`.

### AWS CLI

Per creare una revisione del codice.

Quanto segue `create-code-review` crea una revisione del codice nel mainline ramo di un AWS CodeCommit repository denominato `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"}}}'

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Create code review in Amazon CodeGuru Reviewer](#) nella Amazon CodeGuru Reviewer User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateCodeReview](#)Reference.

## describe-code-review

Il seguente esempio di codice mostra come utilizzare `describe-code-review`.

### AWS CLI

Elenca i dettagli relativi alla revisione del codice.

Di seguito `describe-code-review` sono elencate le informazioni su una revisione del codice nel ramo «mainline» di un AWS CodeCommit repository denominato "»my-repo-name.

```

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"
  }
}
```

Per ulteriori informazioni, consulta [Visualizza i dettagli della revisione del codice](#) nella Guida per l'utente di Amazon CodeGuru Reviewer.

- Per API i dettagli, consulta [DescribeCodeReview AWS CLI Command Reference](#).

## describe-recommendation-feedback

Il seguente esempio di codice mostra come utilizzare `describe-recommendation-feedback`.

### AWS CLI

Per visualizzare informazioni sul feedback su una raccomandazione

Di seguito `describe-recommendation-feedback` vengono visualizzate informazioni sul feedback su un consiglio. Questa raccomandazione ha una sola ThumbsUp reazione.

```
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

```

Output:

```

{
  "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
  }
}

```

Per ulteriori informazioni, consulta [Visualizzare consigli e fornire feedback](#) e [Passaggio 4: fornire feedback](#) nella Amazon CodeGuru Reviewer User Guide.

- Per API i dettagli, consulta [DescribeRecommendationFeedback AWS CLI Command Reference](#).

## describe-repository-association

Il seguente esempio di codice mostra come utilizzare `describe-repository-association`.

AWS CLI

Esempio 1: per restituire informazioni su un'associazione di GitHub repository

L'`describe-repository-association` seguente restituisce informazioni su un'associazione di repository che utilizza un repository GitHub Enterprise e si trova nello stato. Associated

```
aws codeguru-reviewer describe-repository-association \
```



```
--association-arn arn:aws:codeguru-reviewer:us-west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Creare un'associazione di repository GitHub Enterprise Server in Amazon CodeGuru Reviewer nella Amazon CodeGuru Reviewer User Guide](#).

Esempio 2: restituire informazioni su un'associazione di repository non riuscita

L'operazione `describe-repository-association` seguente restituisce informazioni su un'associazione di repository che utilizza un repository GitHub Enterprise e si trova nello stato `Failed`

```
aws codeguru-reviewer describe-repository-association \  
--association-arn arn:aws:codeguru-reviewer:us-west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Creare un'associazione di repository GitHub Enterprise Server in Amazon CodeGuru Reviewer nella Amazon CodeGuru Reviewer User Guide](#).

Esempio 3: per restituire informazioni su un'associazione di repository dissociante

L'output seguente restituisce informazioni su un'associazione di repository che utilizza un repository GitHub Enterprise e si trova nello stato. `Disassociating`

```

aws codeguru-reviewer describe-repository-association \
  --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Creare un'associazione di repository GitHub Enterprise Server in Amazon CodeGuru Reviewer nella Amazon CodeGuru Reviewer User Guide](#).

- Per API i dettagli, consulta Command [DescribeRepositoryAssociation](#) Reference AWS CLI .

## disassociate-repository

Il seguente esempio di codice mostra come utilizzare `disassociate-repository`.

### AWS CLI

Per dissociare un'associazione di repository

Quanto segue `disassociate-repository` dissocia un'associazione di repository che utilizza un repository. AWS CodeCommit

```
aws codeguru-reviewer disassociate-repository \  
  --association-arn arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Dissociare un repository in CodeGuru Reviewer nella Amazon CodeGuru Reviewer User Guide](#).

- Per i API dettagli, consulta [DisassociateRepository](#) Command Reference.AWS CLI

## list-code-reviews

Il seguente esempio di codice mostra come utilizzare `list-code-reviews`.

### AWS CLI

Per elencare le recensioni di codice create nel tuo AWS account negli ultimi 90 giorni.

L'`list-code-reviews` esempio seguente elenca le revisioni del codice create negli ultimi 90 giorni utilizzando le pull request.

```
aws codeguru-reviewer list-code-reviews \  
  --type PullRequest
```

Output:

```
{  
  "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",  
      "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Visualizza tutte le revisioni del codice](#) nella Amazon CodeGuru Reviewer User Guide.

- Per API i dettagli, consulta [ListCodeReviews AWS CLI Command Reference](#).

## list-recommendation-feedback

Il seguente esempio di codice mostra come utilizzare `list-recommendation-feedback`.

### AWS CLI

Per elencare il feedback dei clienti relativo a un consiglio su un repository associato

Di seguito `list-recommendation-feedback` sono elencati i feedback dei clienti su tutti i consigli relativi alla revisione del codice. Questa revisione del codice contiene un feedback, un "ThumbsUp", da parte di un cliente.

```
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
```

Output:

```
{
  "RecommendationFeedbackSummaries": [
    {
      "RecommendationId":
"3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb",
      "Reactions": [
        "ThumbsUp"
      ],
      "UserId": "aws-user-id"
    }
  ]
}
```

Per ulteriori informazioni, consulta la [Fase 4: Fornire feedback](#) nella Guida per l'utente di Amazon CodeGuru Reviewer.

- Per API i dettagli, consulta [ListRecommendationFeedback AWS CLI Command Reference](#).

## list-recommendations

Il seguente esempio di codice mostra come utilizzare `list-recommendations`.

## AWS CLI

Per elencare i consigli per una revisione completa del codice

L'`list-recommendations`esempio seguente elenca i consigli per completare una revisione del codice. Questa revisione del codice contiene una raccomandazione.

```
aws codeguru-reviewer list-recommendations \  
  --code-review-arn arn:aws:codeguru-reviewer:us-west-2:544120495673:code-  
  review:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta la [Fase 4: Fornire feedback](#) nella Guida per l'utente di Amazon CodeGuru Reviewer.

- Per API i dettagli, consulta [ListRecommendations AWS CLI](#) Command Reference.

## `list-repository-associations`

Il seguente esempio di codice mostra come utilizzare `list-repository-associations`.



## AWS CLI

Per elencare le associazioni di repository presenti nel tuo account AWS

L'`list-repository-associations` seguente restituisce un elenco di oggetti di riepilogo delle associazioni di repository presenti nell'account. È possibile filtrare l'elenco restituito per `ProviderType`, `NameState`, `eOwner`.

```
aws codeguru-reviewer list-repository-associations
```

Output:

```
{
  "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"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Visualizza tutte le associazioni di repository in CodeGuru Reviewer nella Amazon CodeGuru Reviewer User Guide](#).

- Per API i dettagli, consulta Command [ListRepositoryAssociations](#) Reference AWS CLI .

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

## AWS CLI

Per elencare i tag in un repository associato

Di seguito `list-tags-for-resource` sono elencati i tag presenti in un repository associato. Questo repository associato ha due tag.

```
aws codeguru-reviewer list-tags-for-resource \  
  --resource-arn arn:aws:codeguru-reviewer:us-  
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "Tags": {  
    "Status": "Secret",  
    "Team": "Saanvi"  
  }  
}
```

Per ulteriori informazioni, consulta [Visualizza tag per un repository associato a CodeGuru Reviewer \(AWS CLI\)](#) nella Amazon CodeGuru Reviewer User Guide.

- Per API i dettagli, consulta Command [ListTagsForResource](#) Reference AWS CLI .

## put-recommendation-feedback

Il seguente esempio di codice mostra come utilizzare `put-recommendation-feedback`.

### AWS CLI

Per aggiungere un consiglio a una revisione del codice

Di seguito viene `put-recommendation-feedback` fornita una ThumbsUp raccomandazione per la revisione del codice.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la [Fase 4: Fornire feedback](#) nella Guida per l'utente di Amazon CodeGuru Reviewer.

- Per API i dettagli, consulta [PutRecommendationFeedback AWS CLI](#) Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere un tag a un repository associato

Quanto segue `tag-resource` aggiunge due tag a un repository associato

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere un tag a un repository associato a CodeGuru Reviewer \(AWS CLI\)](#) e [Aggiungere o aggiornare tag per un repository associato a CodeGuru Reviewer \(\)AWS CLI nella Amazon CodeGuru Reviewer User Guide](#).

- Per i API dettagli, consulta Command Reference. [TagResource](#) AWS CLI

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da un repository associato

Quanto segue `untag-resource` rimuove due tag con le chiavi «Secret» e «Team» da un repository associato.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Rimuovere i tag da un repository associato a CodeGuru Reviewer \(AWS CLI\)](#) nella Amazon CodeGuru Reviewer User Guide.

- Per API i dettagli, consulta Command [UntagResource](#)Reference AWS CLI .

## CodePipeline esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with CodePipeline.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **acknowledge-job**

Il seguente esempio di codice mostra come utilizzare `acknowledge-job`.

AWS CLI

Per recuperare informazioni su un lavoro specifico

Questo esempio restituisce informazioni su un lavoro specifico, incluso lo stato di quel lavoro, se esiste. Viene utilizzato solo per gli addetti ai lavori e le azioni personalizzate. Per determinare il valore di nonce e l'ID del lavoro, usa `aws poll-for-jobs codepipeline`.

Comando:

```
aws codepipeline acknowledge-job --job-id f4f4ff82-2d11-EXAMPLE --nonce 3
```

Output:

```
{  
  "status": "InProgress"  
}
```

- Per API i dettagli, vedere [AcknowledgeJob](#) in AWS CLI Command Reference.

## create-custom-action-type

Il seguente esempio di codice mostra come utilizzare `create-custom-action-type`.

AWS CLI

Per creare un'azione personalizzata

Questo esempio crea un'azione personalizzata per AWS CodePipeline l'utilizzo di un JSON file già creato (qui denominato `MyCustomAction.json`) che contiene la struttura dell'azione personalizzata. Per ulteriori informazioni sui requisiti per la creazione di un'azione personalizzata, inclusa la struttura del file, consulta la Guida per l'utente. AWS CodePipeline

```
aws codepipeline create-custom-action-type --cli-input-json file://  
MyCustomAction.json
```

Contenuto del JSON file `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  
}  
}
```

Questo comando restituisce la struttura dell'azione personalizzata.

- Per API i dettagli, vedere [CreateCustomActionType](#) in AWS CLI Command Reference.

## create-pipeline

Il seguente esempio di codice mostra come utilizzare `create-pipeline`.

### AWS CLI

Per creare una pipeline

Questo esempio crea una pipeline AWS CodePipeline utilizzando un JSON file già creato (qui denominato `MySecondPipeline.json`) che contiene la struttura della pipeline. Per ulteriori informazioni sui requisiti per la creazione di una pipeline, inclusa la struttura del file, consulta la Guida per l'utente. AWS CodePipeline

Comando:

```
aws codepipeline create-pipeline --cli-input-json file://MySecondPipeline.json
```

## JSON contenuti di esempio del file:

```
{
  "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
}
}
```

#### Output:

This command returns the structure of the pipeline.

- Per API i dettagli, vedere [CreatePipeline](#) in AWS CLI Command Reference.

## delete-custom-action-type

Il seguente esempio di codice mostra come utilizzare `delete-custom-action-type`.

### AWS CLI

Per eliminare un'azione personalizzata

Questo esempio elimina un'azione personalizzata AWS CodePipeline utilizzando un JSON file già creato (qui denominato `DeleteMyCustomAction.json`) che contiene il tipo di azione, il nome del provider e il numero di versione dell'azione da eliminare. Utilizzate il `list-action-types` comando per visualizzare i valori corretti per categoria, versione e provider.

Comando:

```
aws codepipeline delete-custom-action-type --cli-input-json file://DeleteMyCustomAction.json
```

JSON contenuto di esempio del file:

```
{
  "category": "Build",
  "version": "1",
  "provider": "MyJenkinsProviderName"
}
```

Output:

```
None.
```

- Per API i dettagli, vedere [DeleteCustomActionType](#) in AWS CLI Command Reference.

## delete-pipeline

Il seguente esempio di codice mostra come utilizzare `delete-pipeline`.

### AWS CLI

Per eliminare una pipeline

Questo esempio elimina una tubazione denominata `MySecondPipeline` AWS CodePipeline. Utilizzate il comando `list-pipelines` per visualizzare un elenco di pipeline associate al vostro account. AWS

Comando:

```
aws codepipeline delete-pipeline --name MySecondPipeline
```

Output:

```
None.
```

- Per i API dettagli, vedere [DeletePipeline](#) in Command Reference. AWS CLI

## delete-webhook

Il seguente esempio di codice mostra come utilizzare `delete-webhook`.

### AWS CLI

Per eliminare un webhook

L'`delete-webhook` esempio seguente elimina un webhook per un'azione di origine della GitHub versione 1. È necessario utilizzare il `deregister-webhook-with-third-party` comando per annullare la registrazione del webhook prima di eliminarlo.

```
aws codepipeline delete-webhook \  
  --name my-webhook
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminare il webhook dalla GitHub fonte nella Guida per l'utente](#).AWS CodePipeline

- Per API i dettagli, vedere [DeleteWebhook](#) in AWS CLI Command Reference.

## deregister-webhook-with-third-party

Il seguente esempio di codice mostra come utilizzare `deregister-webhook-with-third-party`.

### AWS CLI

Per annullare la registrazione di un webhook

L'`deregister-webhook-with-third-party` esempio seguente elimina un webhook per un'azione di origine della versione 1. GitHub Prima di eliminare il webhook, devi annullarne la registrazione.

```
aws codepipeline deregister-webhook-with-third-party \  
  --webhook-name my-webhook
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminare il webhook dalla GitHub fonte nella Guida per l'AWS CodePipeline utente](#).

- Per API i dettagli, vedere [DeregisterWebhookWithThirdParty](#) in AWS CLI Command Reference.

## disable-stage-transition

Il seguente esempio di codice mostra come utilizzare `disable-stage-transition`.

### AWS CLI

Per disabilitare una transizione verso una fase di una pipeline

Questo esempio disabilita le transizioni allo stadio Beta della MyFirstPipeline pipeline in. AWS CodePipeline

Comando:

```
aws codepipeline disable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta --transition-type Inbound
```

Output:

```
None .
```

- Per API i dettagli, consultate Command [DisableStageTransition](#) Reference AWS CLI .

## enable-stage-transition

Il seguente esempio di codice mostra come utilizzare `enable-stage-transition`.

### AWS CLI

Per abilitare la transizione verso una fase di una pipeline

Questo esempio abilita le transizioni allo stadio Beta della MyFirstPipeline pipeline in. AWS CodePipeline

Comando:

```
aws codepipeline enable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta --transition-type Inbound
```

Output:

```
None .
```

- Per API i dettagli, vedere [EnableStageTransition](#) in AWS CLI Command Reference.

## get-job-details

Il seguente esempio di codice mostra come utilizzare `get-job-details`.

### AWS CLI

Per ottenere i dettagli di un lavoro

Questo esempio restituisce dettagli su un lavoro il cui ID è rappresentato da `f4f4ff82-2d11-EXAMPLE`. Questo comando viene utilizzato solo per azioni personalizzate. Quando viene chiamato questo comando, AWS CodePipeline restituisce le credenziali temporanee per il bucket Amazon S3 utilizzato per archiviare gli artefatti per la pipeline, se necessario per l'azione personalizzata. Questo comando restituirà anche tutti i valori segreti definiti per l'azione, se ne sono definiti.

Comando:

```
aws codepipeline get-job-details --job-id f4f4ff82-2d11-EXAMPLE
```

Output:

```
{
  "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"
}
}

```

- Per API i dettagli, vedere [GetJobDetails](#) in AWS CLI Command Reference.

## get-pipeline-state

Il seguente esempio di codice mostra come utilizzare `get-pipeline-state`.

## AWS CLI

Per ottenere informazioni sullo stato di una pipeline

Questo esempio restituisce lo stato più recente di una pipeline denominata. `MyFirstPipeline`

Comando:

```
aws codepipeline get-pipeline-state --name MyFirstPipeline
```

Output:

```
{
  "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
}
```

- Per API i dettagli, vedere [GetPipelineState](#) in AWS CLI Command Reference.

## get-pipeline

Il seguente esempio di codice mostra come utilizzare `get-pipeline`.

### AWS CLI

Per visualizzare la struttura di una pipeline

Questo esempio restituisce la struttura di una tubazione denominata. `MyFirstPipeline`

Comando:

```
aws codepipeline get-pipeline --name MyFirstPipeline
```

Output:

```
{
  "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
}
}
```

- Per API i dettagli, vedere [GetPipeline](#) in AWS CLI Command Reference.

## list-action-executions

Il seguente esempio di codice mostra come utilizzare `list-action-executions`.

### AWS CLI

Per elencare le esecuzioni di azioni

L'`list-action-executions` esempio seguente visualizza i dettagli di esecuzione dell'azione per una pipeline, come l'ID di esecuzione dell'azione, gli artefatti di input, gli artefatti di output, il risultato dell'esecuzione e lo stato.

```
aws codepipeline list-action-executions \
  --pipeline-name myPipeline
```

Output:

```
{
  "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":
"11111111ad99dcd35914c00b7f7bea13995EXAMPLE",
        "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"
    }
}
},
```

. . . .

Per ulteriori informazioni, vedete [View action executions \(\) CLI](#) nella Guida per l'utente.AWS CodePipeline

- Per API i dettagli, vedere [ListActionExecutions](#)in AWS CLI Command Reference.

## list-action-types

Il seguente esempio di codice mostra come utilizzare `list-action-types`.

### AWS CLI

Per visualizzare i tipi di azione disponibili

Utilizzato da solo, il `list-action-types` comando restituisce la struttura di tutte le azioni disponibili per l' AWS account. Questo esempio utilizza l'`action-owner-filter` opzione `--` per restituire solo azioni personalizzate.

Comando:

```
aws codepipeline list-action-types --action-owner-filter Custom
```

Output:

```
{
  "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}"
    }
}
```

```
]
}
```

- Per API i dettagli, vedere [ListActionTypes](#) in AWS CLI Command Reference.

## list-pipeline-executions

Il seguente esempio di codice mostra come utilizzare `list-pipeline-executions`.

### AWS CLI

Per visualizzare la cronologia di esecuzione della pipeline

L'`list-pipeline-executions` esempio seguente mostra la cronologia di esecuzione della pipeline per una pipeline nel tuo account. AWS

```
aws codepipeline list-pipeline-executions \
  --pipeline-name MyPipeline
```

Output:

```
{
  "pipelineExecutionSummaries": [
    {
      "lastUpdateTime": 1496380678.648,
      "pipelineExecutionId": "7cf7f7cb-3137-539g-j458-d7eu3EXAMPLE",
      "startTime": 1496380258.243,
      "status": "Succeeded"
    },
    {
      "lastUpdateTime": 1496591045.634,
      "pipelineExecutionId": "3137f7cb-8d494hj4-039j-d84l-d7eu3EXAMPLE",
      "startTime": 1496590401.222,
      "status": "Succeeded"
    },
    {
      "lastUpdateTime": 1496946071.6456,
      "pipelineExecutionId": "4992f7jf-7cf7-913k-k334-d7eu3EXAMPLE",
      "startTime": 1496945471.5645,
      "status": "Succeeded"
    }
  ]
}
```

```
}
```

Per ulteriori informazioni, consulta [Visualizza la cronologia di esecuzione nella Guida](#) per l'AWS CodePipeline utente.

- Per API i dettagli, vedere [ListPipelineExecutions](#) in AWS CLI Command Reference.

## list-pipelines

Il seguente esempio di codice mostra come utilizzare `list-pipelines`.

### AWS CLI

Per visualizzare un elenco di condotte

Questo esempio elenca tutte le AWS CodePipeline pipeline associate all'account dell' AWS utente.

Comando:

```
aws codepipeline list-pipelines
```

Output:

```
{
  "pipelines": [
    {
      "updated": 1439504274.641,
      "version": 1,
      "name": "MyFirstPipeline",
      "created": 1439504274.641
    },
    {
      "updated": 1436461837.992,
      "version": 2,
      "name": "MySecondPipeline",
      "created": 1436460801.381
    }
  ]
}
```

- Per API i dettagli, vedere [ListPipelines](#) in AWS CLI Command Reference.



## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag

L'`list-tags-for-resource` esempio seguente recupera un elenco di tutti i tag allegati alla risorsa pipeline specificata.

```
aws codepipeline list-tags-for-resource \  
  --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline
```

Output:

```
{  
  "tags": {  
    "Project": "ProjectA",  
    "IscontainerBased": "true"  
  }  
}
```

Per ulteriori informazioni, vedete [View tags for a pipeline \(CLI\)](#) nella Guida per l'AWS CodePipeline utente.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## list-webhooks

Il seguente esempio di codice mostra come utilizzare `list-webhooks`.

### AWS CLI

Per elencare i webhook

L'`list-webhooks` esempio seguente recupera un elenco di tutti i tag allegati alla risorsa pipeline specificata.

```
aws codepipeline list-webhooks \  
  --endpoint-url "https://codepipeline.eu-central-1.amazonaws.com" \  
  --region "eu-central-1"
```

**Output:**

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Elenca i webhook presenti nel tuo account nella Guida](#) per l'utente.AWS CodePipeline

- Per API i dettagli, vedere [ListWebhooks](#) in AWS CLI Command Reference.

**poll-for-jobs**

Il seguente esempio di codice mostra come utilizzare `poll-for-jobs`.

**AWS CLI**

Per visualizzare tutte le offerte di lavoro disponibili

Questo esempio restituisce informazioni su qualsiasi mansione su cui un lavoratore può agire. Questo esempio utilizza un JSON file predefinito (`MyActionTypeInfo.json`) per fornire informazioni sul tipo di azione per cui il job worker elabora i job. Questo comando viene utilizzato solo per

azioni personalizzate. Quando viene chiamato questo comando, AWS CodePipeline restituisce le credenziali temporanee per il bucket Amazon S3 utilizzato per archiviare gli artefatti per la pipeline. Questo comando restituirà anche tutti i valori segreti definiti per l'azione, se ne sono definiti.

Comando:

```
aws codepipeline poll-for-jobs --cli-input-json file://MyActionTypeInfo.json
```

JSON contenuto di esempio del file:

```
{
  "actionTypeId": {
    "category": "Test",
    "owner": "Custom",
    "provider": "MyJenkinsProviderName",
    "version": "1"
  },
  "maxBatchSize": 5,
  "queryParam": {
    "ProjectName": "MyJenkinsTestProject"
  }
}
```

Output:

```
{
  "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":
    "fICcQD6m7oRw0uX0jANBqkqhkiG9w0BAQUFADCBiDELMaKGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwd
+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/
f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/
MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpEIbb30hjZnzcVQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQ
+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": "ef66c259-64f9-EXAMPLE",
  "nonce": "3"
}
]
```

- Per API i dettagli, vedere [PollForJobs](#) in AWS CLI Command Reference.

## put-webhook

Il seguente esempio di codice mostra come utilizzare `put-webhook`.

### AWS CLI

Per creare un webhook

L'esempio seguente crea un webhook per un'azione di origine della GitHub versione 1. Dopo aver creato il webhook, è necessario utilizzare il comando `register-webhook-with-third-party` per registrarlo.

```
aws codepipeline put-webhook \  
  --cli-input-json file://webhook_json.json \  
  --region "eu-central-1"
```

Contenuto di `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"  
    }  
  }  
}
```

Output:

```
{
```

```
    "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"
      }
    ]
  }
}
```

Per ulteriori informazioni, consulta [Creare un webhook per una GitHub fonte nella Guida per l'AWS CodePipeline utente](#).

- Per API i dettagli, vedere [PutWebhook](#) in AWS CLI Command Reference.

## retry-stage-execution

Il seguente esempio di codice mostra come utilizzare `retry-stage-execution`.

### AWS CLI

Per riprovare un'azione fallita

L'`retry-stage-execution` esempio seguente riprova una fase in cui è stata eseguita un'azione fallita.

```
aws codepipeline retry-stage-execution \  
  --pipeline-name MyPipeline \  
  --stage-name Deploy \  
  --pipeline-execution-id b59babff-5f34-EXAMPLE \  
  --retry-mode FAILED_ACTIONS
```

Output:

```
{  
  "pipelineExecutionId": "b59babff-5f34-EXAMPLE"  
}
```

Per ulteriori informazioni, consulta [Retry failed actions \(CLI\) nella Guida](#) per l'AWS CodePipeline utente.

- Per API i dettagli, vedere [RetryStageExecution](#) in AWS CLI Command Reference.

## start-pipeline-execution

Il seguente esempio di codice mostra come utilizzare `start-pipeline-execution`.

AWS CLI

Per eseguire la revisione più recente tramite una pipeline

Questo esempio esegue l'ultima revisione presente nella fase di origine di una pipeline tramite la pipeline denominata "». MyFirstPipeline

Comando:

```
aws codepipeline start-pipeline-execution --name MyFirstPipeline
```

Output:

```
{  
  "pipelineExecutionId": "3137f7cb-7cf7-EXAMPLE"  
}
```

- Per API i dettagli, vedere [StartPipelineExecution](#) in AWS CLI Command Reference.

## stop-pipeline-execution

Il seguente esempio di codice mostra come utilizzare `stop-pipeline-execution`.

### AWS CLI

Per interrompere l'esecuzione di una pipeline

Per impostazione predefinita `stop-pipeline-execution`, l'esempio seguente attende il completamento delle azioni in corso e quindi arresta la pipeline. Non è possibile scegliere di interrompere e attendere se l'esecuzione è già in uno stato di arresto. È possibile scegliere di interrompere e abbandonare un'esecuzione già in uno stato di arresto.

```
aws codepipeline stop-pipeline-execution \  
  --pipeline-name MyFirstPipeline \  
  --pipeline-execution-id d-EXAMPLE \  
  --reason "Stopping pipeline after the build action is done"
```

Questo comando non restituisce alcun output.

Per ulteriori informazioni, vedete [Stop a pipeline execution \(CLI\)](#) nella Guida per l'utente AWS CodePipeline

- Per API i dettagli, vedere [StopPipelineExecution](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare una risorsa

L'`tag-resource` esempio seguente associa un set di tag forniti a una pipeline. Utilizzate questo comando per aggiungere o modificare i tag.

```
aws codepipeline tag-resource \  
  --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \  
  --tags key=Project,value=ProjectA key=IscontainerBased,value=true
```

Questo comando non produce alcun output.



Per ulteriori informazioni, vedete [Aggiungere tag a una pipeline \(CLI\)](#) nella Guida per l'AWS CodePipeline utente.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i AWS tag da una risorsa di connessione

L'`untag-resource` esempio seguente rimuove un tag dalla risorsa specificata.

```
aws codepipeline untag-resource \  
  --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \  
  --tag-keys Project IscontainerBased
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Remove tags from a pipeline \(CLI\)](#) nella Guida per l'AWS CodePipeline utente.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-pipeline

Il seguente esempio di codice mostra come utilizzare `update-pipeline`.

### AWS CLI

Per aggiornare la struttura di una pipeline

Questo esempio utilizza il comando `update-pipeline` con l'argomento `--cli-input-json`. Questo esempio utilizza un JSON file predefinito (`MyFirstPipeline.json`) per aggiornare la struttura di una pipeline. AWS CodePipeline riconosce il nome della pipeline contenuto nel JSON file, quindi applica tutte le modifiche apportate ai campi modificati nella struttura della pipeline per aggiornare la pipeline.

Utilizzate le seguenti linee guida per creare il file predefinito: JSON

Se state lavorando con una struttura di pipeline recuperata utilizzando il comando `get-pipeline`, dovete rimuovere la sezione dei metadati dalla struttura della pipeline nel JSON file (le righe «`metadata`»: `{}` e i campi «`created`», «`pipeline`» e «`updated`» all'interno). Il nome della pipeline ARN non può essere modificato.

Comando:

```
aws codepipeline update-pipeline --cli-input-json file://MyFirstPipeline.json
```

JSONContenuto del file di esempio:

```
{
  "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
}
}

```

### Output:

```

{
  "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
}
}
```

- Per API i dettagli, vedere [UpdatePipeline](#) in AWS CLI Command Reference.

## AWS CodeStar Esempi di notifiche utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS CodeStar Notifications.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **create-notification-rule**

Il seguente esempio di codice mostra come utilizzare `create-notification-rule`.

#### AWS CLI

Per creare una regola di notifica

L'create-notification-ruleesempio seguente utilizza un JSON file denominato `rule.json` per creare una regola di notifica denominata `MyNotificationRule` per un repository denominato `MyDemoRepo` nell' AWS account specificato. Le notifiche con il tipo di FULL dettaglio vengono inviate all'SNSargomento Amazon di destinazione specificato quando vengono creati rami e tag.

```
aws codestar-notifications create-notification-rule \  
--cli-input-json file://rule.json
```

Contenuto di `rule.json`.

```
{  
  "Name": "MyNotificationRule",  
  "EventTypeId": "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"  
}
```

Output:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
}
```

Per ulteriori informazioni, consulta [Creare una regola di notifica](#) nella Guida per l'utente della console AWS Developer Tools.

- Per API i dettagli, consulta [CreateNotificationRule AWS CLI Command Reference](#).

## delete-notification-rule

Il seguente esempio di codice mostra come utilizzare `delete-notification-rule`.

### AWS CLI

Per eliminare una regola di notifica

L'`delete-notification-rule` esempio seguente elimina la regola di notifica specificata.

```
aws codestar-notifications delete-notification-rule \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE
```

Output:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
}
```

Per ulteriori informazioni, consulta [Eliminare una regola di notifica](#) nella Guida per l'utente della console AWS Developer Tools.

- Per API i dettagli, consulta [DeleteNotificationRule AWS CLI Command Reference](#).

## delete-target

Il seguente esempio di codice mostra come utilizzare `delete-target`.

### AWS CLI

Per eliminare un obiettivo di una regola di notifica

L'`delete-target` esempio seguente rimuove la destinazione specificata da tutte le regole di notifica configurate per utilizzarla come destinazione, quindi elimina la destinazione.

```
aws codestar-notifications delete-target \  
  --target-address arn:aws:sns:us-east-1:123456789012:MyNotificationTopic \  
  --force-unsubscribe-all
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminare un oggetto con una regola di notifica](#) nella Guida per l'utente della console AWS Developer Tools.

- Per API i dettagli, consulta [DeleteTarget AWS CLI Command Reference](#).

## describe-notification-rule

Il seguente esempio di codice mostra come utilizzare `describe-notification-rule`.

### AWS CLI

Per recuperare i dettagli di una regola di notifica

L'esempio seguente recupera i dettagli della regola di notifica specificata.

```
aws codestar-notifications describe-notification-rule \  
  --arn arn:aws:codestar-notifications:us-west-2:123456789012:notificationrule/  
dc82df7a-EXAMPLE
```

Output:

```
{  
  "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"
}
```

Per ulteriori informazioni, consulta [Visualizza le regole di notifica](#) nella Guida per l'utente della console AWS Developer Tools.

- Per API i dettagli, consulta [DescribeNotificationRule AWS CLI Command Reference](#).

## list-event-types

Il seguente esempio di codice mostra come utilizzare `list-event-types`.

### AWS CLI

Per ottenere un elenco di tipi di eventi per una regola di notifica

L'`list-event-types` esempio seguente recupera un elenco filtrato di tutti i tipi di eventi di notifica disponibili per CodeDeploy le applicazioni. Se invece non si utilizza alcun filtro, il comando restituisce tutti i tipi di eventi di notifica per tutti i tipi di risorse.

```
aws codestar-notifications list-event-types \
  --filters Name=SERVICE_NAME,Value=CodeDeploy
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Creare una regola di notifica](#) nella Guida per l'utente della console AWS Developer Tools.

- Per API i dettagli, consulta [ListEventTypes AWS CLI Command Reference](#).

## list-notification-rules

Il seguente esempio di codice mostra come utilizzare `list-notification-rules`.

### AWS CLI

Per recuperare un elenco di regole di notifica

L'`list-notification-rules` esempio seguente recupera un elenco di tutte le regole di notifica nella regione specificata AWS .

```
aws codestar-notifications list-notification-rules --region us-east-1
```

Output:

```
{
  "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"
    }
  ]
}
```

```
]
}
```

Per ulteriori informazioni, consulta [Visualizza le regole di notifica](#) nella Guida per l'utente della console AWS Developer Tools.

- Per API i dettagli, consulta [ListNotificationRules AWS CLI Command Reference](#).

## **list-tags-for-resource**

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per ottenere un elenco di tag allegati a una regola di notifica

L'`list-tags-for-resource` esempio seguente recupera un elenco di tutti i tag allegati alla regola di notifica specificata. In questo esempio, alla regola di notifica non sono attualmente associati tag.

```
aws codestar-notifications list-tags-for-resource \
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/fe1efd35-EXAMPLE
```

Output:

```
{
  "Tags": {}
}
```

Per ulteriori informazioni, consulta [Creare una regola di notifica](#) nella Guida per l'utente della console AWS Developer Tools.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## **list-targets**

Il seguente esempio di codice mostra come utilizzare `list-targets`.

### AWS CLI

Per recuperare un elenco di obiettivi delle regole di notifica

L'`list-targets` seguente recupera un elenco di tutti gli obiettivi delle regole di notifica nella regione specificata AWS .

```
aws codestar-notifications list-targets \
  --region us-east-1
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Visualizza gli obiettivi delle regole di notifica](#) nella Guida per l'utente della console AWS Developer Tools.

- Per API i dettagli, vedere [ListTargets](#) in AWS CLI Command Reference.

## subscribe

Il seguente esempio di codice mostra come utilizzare `subscribe`.

### AWS CLI

Per aggiungere un obiettivo a una regola di notifica

L'`subscribe` seguente aggiunge un SNS argomento Amazon come destinazione per la regola di notifica specificata.

```
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
```

Output:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
}
```

Per ulteriori informazioni, consulta [Aggiungere o rimuovere un SNS argomento Amazon come obiettivo per una regola di notifica](#) nella Guida per l'utente della console AWS Developer Tools.

- Per API i dettagli, consulta [Subscribe](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere un tag a una regola di notifica

L'`tag-resource` esempio seguente aggiunge un tag con il nome della chiave `Team` e il valore di `Li_Juan` alla regola di notifica specificata.

```
aws codestar-notifications tag-resource \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
fe1efd35-EXAMPLE \  
  --tags Team=Li_Juan
```

Output:

```
{  
  "Tags": {  
    "Team": "Li_Juan"  
  }  
}
```

Per ulteriori informazioni, consulta [Creare una regola di notifica](#) nella Guida per l'utente della console AWS Developer Tools.

- Per API i dettagli, consulta [TagResource AWS CLI Command Reference](#).

## unsubscribe

Il seguente esempio di codice mostra come utilizzare `unsubscribe`.

### AWS CLI

Per rimuovere un obiettivo da una regola di notifica

L'esempio seguente rimuove un SNS argomento Amazon come destinazione dalla regola di notifica specificata.

```
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
```

Output:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
  "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic"  
}
```

Per ulteriori informazioni, consulta [Aggiungere o rimuovere un SNS argomento Amazon come obiettivo per una regola di notifica](#) nella Guida per l'utente della console AWS Developer Tools.

- Per API i dettagli, consulta [Annullare l'iscrizione](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da una regola di notifica

L'`untag-resource` seguente rimuove il tag con il nome della chiave `Team` dalla regola di notifica specificata.

```
aws codestar-notifications untag-resource \  
  --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
fe1efd35-EXAMPLE \  
  --tag-keys Team
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Modifica di una regola di notifica](#) nella Guida per l'utente della console AWS Developer Tools.

- Per API i dettagli, consulta [UntagResource AWS CLI Command Reference](#).

## update-notification-rule

Il seguente esempio di codice mostra come utilizzare `update-notification-rule`.

### AWS CLI

Per aggiornare una regola di notifica

L'`update-notification-rule` seguente aggiorna una regola di notifica denominata `MyNotificationRule` nell' AWS account `123456789012` utilizzando un JSON file denominato `update.json`.

```
aws codestar-notifications update-notification-rule \  
  --cli-input-json file://update.json
```

Contenuto di `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"  
  }  
}
```

Output:

```
{  
  "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/  
dc82df7a-EXAMPLE"  
}
```

Per ulteriori informazioni, consulta [Modificare una regola di notifica](#) nella Guida per l'utente della console AWS Developer Tools.

- Per API i dettagli, consulta [UpdateNotificationRule AWS CLI Command Reference](#).

## CodeConnections esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with CodeConnections.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-connection**

Il seguente esempio di codice mostra come utilizzare `create-connection`.



## AWS CLI

Per creare una connessione

L'`create-connection` esempio seguente mostra come creare una connessione a un repository di terze parti. Questo esempio crea una connessione in cui il provider di terze parti è Bitbucket.

Per impostazione predefinita, una connessione creata tramite AWS CLI o AWS CloudFormation è in sospeso. Dopo aver creato una connessione con CLI o AWS CloudFormation, usa la console per modificare la connessione e renderne lo stato Disponibile.

```
aws codestar-connections create-connection \  
  --provider-type Bitbucket \  
  --connection-name MyConnection
```

Output:

```
{  
  "ConnectionArn": "arn:aws:codestar-connections:us-  
east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"  
}
```

Per ulteriori informazioni, consulta [Creare una connessione](#) nella Guida per l'utente della console Developer Tools.

- Per API i dettagli, consulta [CreateConnection AWS CLI Command Reference](#).

## **create-host**

Il seguente esempio di codice mostra come utilizzare `create-host`.

## AWS CLI

Per creare un host

L'`create-host` esempio seguente mostra come creare un host che rappresenti l'endpoint per l'infrastruttura in cui è installato il provider di terze parti. Questo esempio crea un host in cui il provider installato da terze parti è GitHub Enterprise Server.

Per impostazione predefinita, un host creato tramite il AWS CLI è in sospeso. Dopo aver creato un host con CLI, utilizza la console o CLI per configurare l'host in modo da renderne disponibile lo stato.

```
aws codestar-connections create-host \  
  --name MyHost \  
  --provider-type GitHubEnterpriseServer \  
  --provider-endpoint "https://my-instance.dev"
```

Output:

```
{  
  "HostArn": "arn:aws:codestar-connections:us-east-1:123456789012:host/My-  
Host-28aef605"  
}
```

Per ulteriori informazioni, consulta [Create a host \(CLI\)](#) nella Guida per l'utente della console Developer Tools.

- Per API i dettagli, consulta [CreateHost AWS CLI Command Reference](#).

## delete-connection

Il seguente esempio di codice mostra come utilizzare `delete-connection`.

### AWS CLI

Per eliminare una connessione

L'`delete-connection` esempio seguente mostra come eliminare una connessione.

```
aws codestar-connections delete-connection \  
  --connection-arn arn:aws:codestar-connections:us-west-2:123456789012:connection/  
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminare una connessione \(CLI\)](#) nella Guida per l'utente della console Developer Tools.

- Per API i dettagli, consulta [DeleteConnection AWS CLI Command Reference](#).

## delete-host

Il seguente esempio di codice mostra come utilizzare `delete-host`.

## AWS CLI

Per eliminare un host

L'`delete-host` esempio seguente mostra come eliminare un host. Per eliminare un host, è necessario eliminare tutte le connessioni associate all'host.

```
aws codestar-connections delete-host \  
  --host-arn "arn:aws:codestar-connections:us-east-1 :123456789012:host/My-  
Host-28aef605"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Delete a host \(CLI\)](#) nella Guida per l'utente della console Developer Tools.

- Per API i dettagli, consulta [DeleteHost AWS CLI Command Reference](#).

## get-connection

Il seguente esempio di codice mostra come utilizzare `get-connection`.

## AWS CLI

Per ottenere informazioni su una connessione

L'`get-connection` esempio seguente mostra i dettagli su una connessione.

```
aws codestar-connections get-connection \  
  --connection-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/  
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

Output:

```
{  
  "Connection": {  
    "ConnectionName": "MyConnection",  
    "ConnectionArn": "arn:aws:codestar-connections:us-  
east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",  
    "ProviderType": "Bitbucket",  
    "OwnerAccountId": "123456789012",  
    "ConnectionStatus": "AVAILABLE"  
  }  
}
```

```
}
```

Per ulteriori informazioni, consulta [Visualizza i dettagli della connessione](#) nella Guida per l'utente della console Developer Tools.

- Per API i dettagli, vedere [GetConnection](#) in AWS CLI Command Reference.

## get-host

Il seguente esempio di codice mostra come utilizzare `get-host`.

### AWS CLI

Per ottenere informazioni su un host

L'`get-host` esempio seguente mostra i dettagli su un host:

```
aws codestar-connections get-host \  
  --host-arn arn:aws:codestar-connections:us-east-1:123456789012:host/  
  MyHost-28aef605
```

Output:

```
{  
  "Name": "MyHost",  
  "Status": "AVAILABLE",  
  "ProviderType": "GitHubEnterpriseServer",  
  "ProviderEndpoint": "https://test-instance-1.dev/"  
}
```

Per ulteriori informazioni, consulta [View host details \(CLI\)](#) nella Guida per l'utente della console Developer Tools.

- Per API i dettagli, vedere [GetHost](#) in AWS CLI Command Reference.

## list-connections

Il seguente esempio di codice mostra come utilizzare `list-connections`.

### AWS CLI

Per elencare le connessioni

L'list-connectionsesempio seguente recupera un elenco di tutte le connessioni nel tuo account per il tipo di provider Bitbucket. :

```
aws codestar-connections list-connections \  
--provider-type Bitbucket \  
--max-results 5 \  
--next-token: next-token
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [List connections \(CLI\)](#) nella Guida per l'utente della console Developer Tools.

- Per API i dettagli, consulta [ListConnections AWS CLI Command Reference](#).

## list-hosts

Il seguente esempio di codice mostra come utilizzare list-hosts.

## AWS CLI

Per elencare gli host

L'`list-hosts` esempio seguente recupera un elenco di tutti gli host del tuo account.

```
aws codestar-connections list-hosts
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [List hosts \(CLI\)](#) nella Guida per l'utente della console Developer Tools.

- Per API i dettagli, consulta [ListHosts AWS CLI Command Reference](#).

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag

L'`list-tags-for-resource` esempio seguente recupera un elenco di tutti i tag allegati alla risorsa di connessione specificata.

```
aws codestar-connections list-tags-for-resource \
  --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

## Output:

```
{
  "Tags": [
    {
      "Key": "Project",
      "Value": "ProjectA"
    },
    {
      "Key": "ReadOnly",
      "Value": "true"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Visualizza i tag per una risorsa di connessioni](#) nella Guida per l'utente della console Developer Tools.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare una risorsa

L'example seguente associa un set di tag forniti a una connessione. Utilizzate questo comando per aggiungere o modificare i tag.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere tag a una risorsa di connessioni](#) nella Guida per l'utente della console Developer Tools.

- Per API i dettagli, consulta [TagResource AWS CLI Command Reference](#).

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i AWS tag da una risorsa di connessione

Quanto segue `untag-resource` rimuove un tag dalla risorsa specificata.

```
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
```

Output:

```
{  
  "Tags": []  
}
```

Per ulteriori informazioni, consulta [Rimuovere tag da una risorsa di connessioni](#) nella Guida per l'utente della console Developer Tools.

- Per API i dettagli, consulta [UntagResource AWS CLI Command Reference](#).

## Esempi di identità di Amazon Cognito utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Cognito Identity. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)



## Azioni

### **create-identity-pool**

Il seguente esempio di codice mostra come utilizzare `create-identity-pool`.

#### AWS CLI

Per creare un pool di identità con il provider del pool di identità Cognito

Questo esempio crea un pool di identità denominato `MyIdentityPool`. Dispone di un provider del pool di identità Cognito. Le identità non autenticate non sono consentite.

Comando:

```
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
```

Output:

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [CreatIdentityPool](#) in AWS CLI Command Reference.

### **delete-identities**

Il seguente esempio di codice mostra come utilizzare `delete-identities`.

## AWS CLI

Per eliminare un pool di identità

Questo esempio elimina un pool di identità.

Comando:

```
aws cognito-identity delete-identity-pool --identity-ids-to-delete "us-west-2:11111111-1111-1111-1111-111111111111"
```

Output:

```
{
  "UnprocessedIdentityIds": []
}
```

- Per API i dettagli, vedere [DeleteIdentities](#) in AWS CLI Command Reference.

## delete-identity-pool

Il seguente esempio di codice mostra come utilizzare `delete-identity-pool`.

### AWS CLI

Per eliminare un pool di identità

Nell'esempio `delete-identity-pool` seguente viene eliminato il pool di identità specificato.

Comando:

```
aws cognito-identity delete-identity-pool \
  --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteIdentityPool](#) in AWS CLI Command Reference.

## describe-identity-pool

Il seguente esempio di codice mostra come utilizzare `describe-identity-pool`.

## AWS CLI

Per descrivere un pool di identità

Questo esempio descrive un pool di identità.

Comando:

```
aws cognito-identity describe-identity-pool --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

Output:

```
{
  "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": "3n4b5urk1ft4f13mg5e62d9ado",
      "ServerSideTokenCheck": false
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeIdentityPool](#) in AWS CLI Command Reference.

## get-identity-pool-roles

Il seguente esempio di codice mostra come utilizzare `get-identity-pool-roles`.

## AWS CLI

Per ottenere i ruoli del pool di identità

Questo esempio ottiene i ruoli del pool di identità.

Comando:

```
aws cognito-identity get-identity-pool-roles --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

## Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [GetIdentityPoolRoles](#) in AWS CLI Command Reference.

## list-identity-pools

Il seguente esempio di codice mostra come utilizzare `list-identity-pools`.

### AWS CLI

Per elencare pool di identità

In questo esempio vengono elencati i pool di identità. Sono elencate un massimo di 20 identità.

Comando:

```
aws cognito-identity list-identity-pools --max-results 20
```

## Output:

```
{
  "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"
    }
]
}
```

- Per API i dettagli, vedere [ListIdentityPools](#) in AWS CLI Command Reference.

## set-identity-pool-roles

Il seguente esempio di codice mostra come utilizzare `set-identity-pool-roles`.

### AWS CLI

Per impostare i ruoli del pool di identità

L'esempio seguente imposta un ruolo del pool di identità.

```
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"
```

- Per API i dettagli, vedere [SetIdentityPoolRoles](#) in AWS CLI Command Reference.

## update-identity-pool

Il seguente esempio di codice mostra come utilizzare `update-identity-pool`.

### AWS CLI

Per aggiornare un pool di identità

Questo esempio aggiorna un pool di identità. Imposta il nome su `MyIdentityPool`. Aggiunge Cognito come provider di identità. Non consente identità non autenticate.

Comando:

```
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_111111111",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

Output:

```
{
  "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_111111111",
      "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
      "ServerSideTokenCheck": false
    }
  ]
}
```

- Per i API dettagli, vedere [UpdateIdentityPool](#) in Command Reference.AWS CLI

## Esempi di Amazon Cognito Identity Provider che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Cognito Identity Provider. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **add-custom-attributes**

Il seguente esempio di codice mostra come utilizzare `add-custom-attributes`.

## AWS CLI

Per aggiungere un attributo personalizzato

Questo esempio aggiunge un attributo personalizzato CustomAttr 1 a un pool di utenti. È di tipo String e richiede un minimo di 1 carattere e un massimo di 15. ma non è obbligatorio.

Comando:

```
aws cognito-idp add-custom-attributes --user-pool-id us-west-2_aaaaaaaa --custom-attributes
Name="CustomAttr1",AttributeDataType="String",DeveloperOnlyAttribute=false,Required=false,S
```

- Per API i dettagli, vedere [AddCustomAttributes](#) in AWS CLI Command Reference.

## admim-disable-user

Il seguente esempio di codice mostra come utilizzare `admim-disable-user`.

## AWS CLI

Per disabilitare un utente

Questo esempio disabilita l'utente `jane@example.com`.

Comando:

```
aws cognito-idp admin-disable-user --user-pool-id us-west-2_aaaaaaaa --
username jane@example.com
```

- Per API i dettagli, vedere [AdminDisableUser](#) in AWS CLI Command Reference.

## admim-enable-user

Il seguente esempio di codice mostra come utilizzare `admim-enable-user`.

## AWS CLI

Per abilitare un utente

Questo esempio abilita il nome utente `jane@example.com`.

Comando:

```
aws cognito-idp admin-enable-user --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

- Per API i dettagli, vedere [AdminEnableUser](#) in AWS CLI Command Reference.

## admin-add-user-to-group

Il seguente esempio di codice mostra come utilizzare `admin-add-user-to-group`.

AWS CLI

Per aggiungere un utente a un gruppo

Questo esempio aggiunge l'utente Jane al gruppo MyGroup.

Comando:

```
aws cognito-idp admin-add-user-to-group --user-pool-id us-west-2_aaaaaaaaa --  
username Jane --group-name MyGroup
```

- Per API i dettagli, vedere [AdminAddUserToGroup](#) in AWS CLI Command Reference.

## admin-confirm-sign-up

Il seguente esempio di codice mostra come utilizzare `admin-confirm-sign-up`.

AWS CLI

Per confermare la registrazione dell'utente

Questo esempio conferma l'utente `jane@example.com`.

Comando:

```
aws cognito-idp admin-confirm-sign-up --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

- Per API i dettagli, vedere [AdminConfirmSignUp](#) in AWS CLI Command Reference.



## admin-create-user

Il seguente esempio di codice mostra come utilizzare `admin-create-user`.

### AWS CLI

Per creare un utente

L'`admin-create-user` seguente crea un utente con le impostazioni specificate: indirizzo e-mail e numero di telefono.

```
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
```

Output:

```
{  
  "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"  
  }  
}
```

- Per API i dettagli, vedere [AdminCreateUser](#) in AWS CLI Command Reference.

## admin-delete-user-attributes

Il seguente esempio di codice mostra come utilizzare `admin-delete-user-attributes`.

### AWS CLI

Per eliminare un attributo utente

Questo esempio elimina un attributo personalizzato CustomAttr 1 per l'utente `diego@example.com`.

Comando:

```
aws cognito-idp admin-delete-user-attributes --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com --user-attribute-names "custom:CustomAttr1"
```

- Per API i dettagli, vedere [AdminDeleteUserAttributes](#) in AWS CLI Command Reference.

## admin-delete-user

Il seguente esempio di codice mostra come utilizzare `admin-delete-user`.

### AWS CLI

Per eliminare un utente

Questo esempio elimina un utente.

Comando:

```
aws cognito-idp admin-delete-user --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com
```

- Per API i dettagli, vedere [AdminDeleteUser](#) in AWS CLI Command Reference.

## admin-forget-device

Il seguente esempio di codice mostra come utilizzare `admin-forget-device`.

## AWS CLI

Dimenticare un dispositivo

Questo esempio dimentica il nome utente del dispositivo jane@example.com

Comando:

```
aws cognito-idp admin-forget-device --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com --device-key us-west-2_abcd_1234-5678
```

- Per API i dettagli, vedere [AdminForgetDevice](#) in AWS CLI Command Reference.

## admin-get-device

Il seguente esempio di codice mostra come utilizzare `admin-get-device`.

### AWS CLI

Per procurarsi un dispositivo

Questo esempio ottiene un dispositivo come nome utente jane@example.com

Comando:

```
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
```

- Per API i dettagli, vedere [AdminGetDevice](#) in AWS CLI Command Reference.

## admin-get-user

Il seguente esempio di codice mostra come utilizzare `admin-get-user`.

### AWS CLI

Per ottenere un utente

Questo esempio ottiene informazioni sul nome utente jane@example.com.

Comando:

```
aws cognito-idp admin-get-user --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

Output:

```
{  
  "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  
}
```

- Per API i dettagli, vedere [AdminGetUser](#) in AWS CLI Command Reference.

## admin-initiate-auth

Il seguente esempio di codice mostra come utilizzare `admin-initiate-auth`.

## AWS CLI

Per avviare l'autorizzazione

Questo esempio avvia l'autorizzazione utilizzando il AUTH flusso ADMIN\_NO\_SRP\_ per il nome utente jane@example.com

Il client deve avere abilitato l'accesso API per l'autenticazione basata sul server (ADMIN\_NO\_).  
SRP AUTH

Usa le informazioni sulla sessione nel valore restituito per chiamare -challenge. admin-respond-to-auth

Comando:

```
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
```

Output:

```
{  
  "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\"}"  
  }  
}
```

- Per API i dettagli, vedere [AdminInitiateAuth](#) in AWS CLI Command Reference.

## admin-list-devices

Il seguente esempio di codice mostra come utilizzare `admin-list-devices`.

## AWS CLI

Per elencare i dispositivi per un utente

Questo esempio elenca i dispositivi per il nome utente `jane@example.com`.

Comando:

```
aws cognito-idp admin-list-devices --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com
```

- Per API i dettagli, vedere [AdminListDevices](#) in AWS CLI Command Reference.

## admin-list-groups-for-user

Il seguente esempio di codice mostra come utilizzare `admin-list-groups-for-user`.

AWS CLI

Per elencare i gruppi per un utente

Questo esempio elenca i gruppi per il nome utente `jane@example.com`.

Comando:

```
aws cognito-idp admin-list-groups-for-user --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com
```

Output:

```
{  
  "Groups": [  
    {  
      "Description": "Sample group",  
      "Precedence": 1,  
      "LastModifiedDate": 1548097827.125,  
      "RoleArn": "arn:aws:iam::111111111111:role/SampleRole",  
      "GroupName": "SampleGroup",  
      "UserPoolId": "us-west-2_aaaaaaaaa",  
      "CreationDate": 1548097827.125  
    }  
  ]  
}
```

- Per API i dettagli, vedere [AdminListGroupForUser](#) in AWS CLI Command Reference.

## admin-list-user-auth-events

Il seguente esempio di codice mostra come utilizzare `admin-list-user-auth-events`.

### AWS CLI

Per elencare gli eventi di autorizzazione per un utente

Questo esempio elenca gli eventi di autorizzazione per il nome utente `diego@example.com`.

Comando:

```
aws cognito-idp admin-list-user-auth-events --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com
```

- Per API i dettagli, vedere [AdminListUserAuthEvents](#) in AWS CLI Command Reference.

## admin-remove-user-from-group

Il seguente esempio di codice mostra come utilizzare `admin-remove-user-from-group`.

### AWS CLI

Per rimuovere un utente da un gruppo

Questo esempio rimuove `jane@example.com` da `SampleGroup`.

Comando:

```
aws cognito-idp admin-remove-user-from-group --user-pool-id us-west-2_aaaaaaaaa --  
username jane@example.com --group-name SampleGroup
```

- Per API i dettagli, vedere [AdminRemoveUserFromGroup](#) in AWS CLI Command Reference.

## admin-reset-user-password

Il seguente esempio di codice mostra come utilizzare `admin-reset-user-password`.

### AWS CLI

Per reimpostare la password di un utente

Questo esempio reimposta la password per `diego@example.com`.

Comando:

```
aws cognito-idp admin-reset-user-password --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com
```

- Per API i dettagli, vedere [AdminResetUserPassword](#) in AWS CLI Command Reference.

## admin-set-user-mfa-preference

Il seguente esempio di codice mostra come utilizzare `admin-set-user-mfa-preference`.

AWS CLI

Per impostare le MFA preferenze dell'utente

Questo esempio imposta la SMS MFA preferenza per il nome utente `diego@example.com`.

Comando:

```
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
```

- Per API i dettagli, vedere [AdminSetUserMfaPreference](#) in AWS CLI Command Reference.

## admin-set-user-settings

Il seguente esempio di codice mostra come utilizzare `admin-set-user-settings`.

AWS CLI

Per configurare le impostazioni utente

Questo esempio imposta la preferenza MFA di consegna per il nome utente `diego@example.com` su EMAIL.

Comando:

```
aws cognito-idp admin-set-user-settings --user-pool-id us-west-2_aaaaaaaaa --  
username diego@example.com --mfa-options DeliveryMedium=EMAIL
```

- Per API i dettagli, vedere [AdminSetUserSettings](#) in AWS CLI Command Reference.



## admin-update-auth-event-feedback

Il seguente esempio di codice mostra come utilizzare `admin-update-auth-event-feedback`.

### AWS CLI

Per fornire un feedback per un evento di autorizzazione

Questo esempio imposta il valore di feedback per un evento di autorizzazione identificato da `event-id` su `Valid`.

Comando:

```
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
```

- Per API i dettagli, vedere [AdminUpdateAuthEventFeedback](#) in AWS CLI Command Reference.

## admin-update-device-status

Il seguente esempio di codice mostra come utilizzare `admin-update-device-status`.

### AWS CLI

Per aggiornare lo stato del dispositivo

Questo esempio imposta lo stato di memorizzazione del dispositivo per il dispositivo identificato da `device-key` su `not_memored`.

Comando:

```
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
```

- Per i API dettagli, vedere in Command Reference. [AdminUpdateDeviceStatus](#) AWS CLI

## admin-update-user-attributes

Il seguente esempio di codice mostra come utilizzare `admin-update-user-attributes`.

## AWS CLI

Per aggiornare gli attributi utente

Questo esempio aggiorna un attributo utente personalizzato CustomAttr 1 per l'utente diego@example.com.

Comando:

```
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"
```

- Per API i dettagli, vedere [AdminUpdateUserAttributes](#) in AWS CLI Command Reference.

## change-password

Il seguente esempio di codice mostra come utilizzare `change-password`.

### AWS CLI

Per modificare una password

Questo esempio modifica una password.

Comando:

```
aws cognito-idp change-password --previous-password OldPassword --proposed-password NewPassword --access-token ACCESS_TOKEN
```

- Per API i dettagli, vedere [ChangePassword](#) in AWS CLI Command Reference.

## confirm-forgot-password

Il seguente esempio di codice mostra come utilizzare `confirm-forgot-password`.

### AWS CLI

Per confermare una password dimenticata

Questo esempio conferma una password dimenticata per il nome utente diego@example.com.

Comando:

```
aws cognito-idp confirm-forgot-password --client-id 3n4b5urk1ft4f13mg5e62d9ado --username=diego@example.com --password PASSWORD --confirmation-code CONF_CODE
```

- Per API i dettagli, vedere [ConfirmForgotPassword](#) in AWS CLI Command Reference.

## confirm-sign-up

Il seguente esempio di codice mostra come utilizzare `confirm-sign-up`.

AWS CLI

Per confermare la registrazione

Questo esempio conferma la registrazione per il nome utente `diego@example.com`.

Comando:

```
aws cognito-idp confirm-sign-up --client-id 3n4b5urk1ft4f13mg5e62d9ado --username=diego@example.com --confirmation-code CONF_CODE
```

- Per API i dettagli, vedere [ConfirmSignUp](#) in AWS CLI Command Reference.

## create-group

Il seguente esempio di codice mostra come utilizzare `create-group`.

AWS CLI

Per creare un gruppo

Questo esempio crea un gruppo con una descrizione.

Comando:

```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyNewGroup --description "New group."
```

Output:

```
{
```

```
"Group": {
  "GroupName": "MyNewGroup",
  "UserPoolId": "us-west-2_aaaaaaaaaa",
  "Description": "New group.",
  "LastModifiedDate": 1548270073.795,
  "CreationDate": 1548270073.795
}
```

Per creare un gruppo con un ruolo e una precedenza

Questo esempio crea un gruppo con una descrizione. Include anche un ruolo e una precedenza.

Comando:

```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaaaa --group-
name MyNewGroupWithRole --description "New group with a role." --role-
arn arn:aws:iam::111111111111:role/MyNewGroupRole --precedence 2
```

Output:

```
{
  "Group": {
    "GroupName": "MyNewGroupWithRole",
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "Description": "New group with a role.",
    "RoleArn": "arn:aws:iam::111111111111:role/MyNewGroupRole",
    "Precedence": 2,
    "LastModifiedDate": 1548270211.761,
    "CreationDate": 1548270211.761
  }
}
```

- Per API i dettagli, vedere [CreateGroup](#) in AWS CLI Command Reference.

## create-user-import-job

Il seguente esempio di codice mostra come utilizzare `create-user-import-job`.

AWS CLI

Per creare un processo di importazione utente

Questo esempio crea un processo di importazione utente denominato MyImportJob.

Per ulteriori informazioni sull'importazione di utenti, vedere Importazione di utenti in pool di utenti da un CSV file.

Comando:

```
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
```

Output:

```
{  
  "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  
  }  
}
```

Carica il file.csv con curl utilizzando il file prefirato: URL

Comando:

```
curl -v -T "PATH_TO_CSV_FILE" -H "x-amz-server-side-encryption:aws:kms"  
"PRE_SIGNED_URL"
```

- Per i API dettagli, consulta [CreateUserImportJob](#) Command Reference.AWS CLI

## create-user-pool-client

Il seguente esempio di codice mostra come utilizzare `create-user-pool-client`.

## AWS CLI

Per creare un client per un pool di utenti

Questo esempio crea un nuovo client con pool di utenti con due flussi di autorizzazione espliciti: USER \_\_ AUTH e PASSWORD ADMIN SRP \_NO\_ . AUTH

Comando:

```
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"
```

Output:

```
{
  "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
  }
}
```

- Per API i dettagli, vedere [CreateUserPoolClient](#) in AWS CLI Command Reference.

## create-user-pool-domain

Il seguente esempio di codice mostra come utilizzare `create-user-pool-domain`.

## AWS CLI

Per creare un dominio con pool di utenti

Questo esempio crea un nuovo dominio di pool di utenti con due flussi di autorizzazione espliciti: USER \_\_ AUTH e ADMIN \_NO\_ PASSWORD \_ . SRP AUTH

Comando:

```
aws cognito-idp create-user-pool-domain --user-pool-id us-west-2_aaaaaaaaa --  
domain my-new-domain
```

- Per i API dettagli, vedere in Command Reference. [CreateUserPoolDomain](#)AWS CLI

## create-user-pool

Il seguente esempio di codice mostra come utilizzare `create-user-pool`.

### AWS CLI

Creazione di un pool di utenti con configurazione minima

Questo esempio crea un pool di utenti denominato `MyUserPool` utilizzando valori predefiniti. Non ci sono attributi obbligatori né client di applicazioni. MFA e la protezione avanzata è disabilitata.

Comando:

```
aws cognito-idp create-user-pool --pool-name MyUserPool
```

Output:

```
{  
  "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": {}
}
```

```
}
```

## Creazione di un pool di utenti con due attributi richiesti

Questo esempio crea un pool di utenti `MyUserPool`. Il pool è configurato per accettare l'e-mail come attributo del nome utente. Inoltre, imposta l'indirizzo e-mail di origine su un indirizzo convalidato utilizzando Amazon Simple Email Service (Amazon SES).

Comando:

```
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"
```

Output:

```
{
  "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": {}
}
}
```

- Per API i dettagli, vedere [CreateUserPool](#) in AWS CLI Command Reference.

## delete-group

Il seguente esempio di codice mostra come utilizzare `delete-group`.

### AWS CLI

Per eliminare un gruppo

Questo esempio elimina un gruppo.

Comando:

```
aws cognito-idp delete-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroupName
```

- Per API i dettagli, vedere [DeleteGroup](#) in AWS CLI Command Reference.

## delete-identity-provider

Il seguente esempio di codice mostra come utilizzare `delete-identity-provider`.

### AWS CLI

Per eliminare un provider di identità

Questo esempio elimina un provider di identità.

Comando:

```
aws cognito-idp delete-identity-provider --user-pool-id us-west-2_aaaaaaaaa --provider-name Facebook
```

- Per API i dettagli, vedere [DeleteIdentityProvider](#) in AWS CLI Command Reference.

## delete-resource-server

Il seguente esempio di codice mostra come utilizzare `delete-resource-server`.

## AWS CLI

Per eliminare un server di risorse

Questo esempio elimina un server di risorse denominato `weather.example.com`.

Comando:

```
aws cognito-idp delete-resource-server --user-pool-id us-west-2_aaaaaaaaa --  
identifier weather.example.com
```

- Per i API dettagli, vedere in Command Reference. [DeleteResourceServer](#) AWS CLI

## **delete-user-attributes**

Il seguente esempio di codice mostra come utilizzare `delete-user-attributes`.

### AWS CLI

Per eliminare gli attributi utente

Questo esempio elimina l'attributo utente "FAVORITE\_ANIMAL».

Comando:

```
aws cognito-idp delete-user-attributes --access-token ACCESS_TOKEN --user-attribute-  
names "FAVORITE_ANIMAL"
```

- Per API i dettagli, vedere [DeleteUserAttributes](#) in AWS CLI Command Reference.

## **delete-user-pool-client**

Il seguente esempio di codice mostra come utilizzare `delete-user-pool-client`.

### AWS CLI

Per eliminare un client del pool di utenti

Questo esempio elimina un client del pool di utenti.

Comando:

```
aws cognito-idp delete-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --client-id 38fjsnc484p94kpbsnet7mpld0
```

- Per API i dettagli, vedere [DeleteUserPoolClient](#) in AWS CLI Command Reference.

## delete-user-pool-domain

Il seguente esempio di codice mostra come utilizzare `delete-user-pool-domain`.

### AWS CLI

Per eliminare un dominio del pool di utenti

L'esempio seguente elimina un dominio del pool di utenti denominato `my-domain`

```
aws cognito-idp delete-user-pool-domain \  
  --user-pool-id us-west-2_aaaaaaaaa \  
  --domain my-domain
```

- Per API i dettagli, vedere [DeleteUserPoolDomain](#) in AWS CLI Command Reference.

## delete-user-pool

Il seguente esempio di codice mostra come utilizzare `delete-user-pool`.

### AWS CLI

Per eliminare un pool di utenti

Questo esempio elimina un pool di utenti utilizzando l'id del pool di utenti, `us-west-2_aaaaaaaaa`.

Comando:

```
aws cognito-idp delete-user-pool --user-pool-id us-west-2_aaaaaaaaa
```

- API Per i [DeleteUserPool](#) dettagli AWS CLI , vedere in Command Reference.

## delete-user

Il seguente esempio di codice mostra come utilizzare `delete-user`.

## AWS CLI

Per eliminare un utente

Questo esempio elimina un utente.

Comando:

```
aws cognito-idp delete-user --access-token ACCESS_TOKEN
```

- Per API i dettagli, vedere [DeleteUser](#) in AWS CLI Command Reference.

## describe-identity-provider

Il seguente esempio di codice mostra come utilizzare `describe-identity-provider`.

## AWS CLI

Per descrivere un provider di identità

Questo esempio descrive un provider di identità chiamato Facebook.

Comando:

```
aws cognito-idp describe-identity-provider --user-pool-id us-west-2_aaaaaaaaa --  
provider-name Facebook
```

Output:

```
{  
  "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
  }
}
```

- Per API i dettagli, vedere [DescribeIdentityProvider](#) in AWS CLI Command Reference.

## describe-resource-server

Il seguente esempio di codice mostra come utilizzare `describe-resource-server`.

### AWS CLI

Per descrivere un server di risorse

Questo esempio descrive il server di risorse `weather.example.com`.

Comando:

```
aws cognito-idp describe-resource-server --user-pool-id us-west-2_aaaaaaaaaa --
identifier weather.example.com
```

Output:

```
{
  "ResourceServer": {
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "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"
    }
  ]
}
```

- Per i API dettagli, vedere [DescribeResourceServer](#) in Command Reference.AWS CLI

## describe-risk-configuration

Il seguente esempio di codice mostra come utilizzare `describe-risk-configuration`.

### AWS CLI

Per descrivere una configurazione di rischio

Questo esempio descrive la configurazione del rischio associata al pool `us-west-2_aaaaaaaaaa`.

Comando:

```
aws cognito-idp describe-risk-configuration --user-pool-id us-west-2_aaaaaaaaaa
```

Output:

```
{
  "RiskConfiguration": {
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "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</
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>",
    }
}

```



```
aws cognito-idp describe-user-import-job --user-pool-id us-west-2_aaaaaaaaaa --job-id import-TZqNQvDRnW
```

Output:

```
{
  "UserImportJob": {
    "JobName": "import-Test1",
    "JobId": "import-TZqNQvDRnW",
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "PreSignedUrl": "PRE_SIGNED URL",
    "CreationDate": 1548271708.512,
    "Status": "Created",
    "CloudWatchLogsRoleArn": "arn:aws:iam::111111111111:role/CognitoCloudWatchLogsRole",
    "ImportedUsers": 0,
    "SkippedUsers": 0,
    "FailedUsers": 0
  }
}
```

- Per API i dettagli, vedere [DescribeUserImportJob](#) in AWS CLI Command Reference.

## describe-user-pool-client

Il seguente esempio di codice mostra come utilizzare `describe-user-pool-client`.

### AWS CLI

Per descrivere un client per un pool di utenti

Questo esempio descrive un client con pool di utenti.

Comando:

```
aws cognito-idp describe-user-pool-client --user-pool-id us-west-2_aaaaaaaaaa --client-id 38fjsnc484p94kpbsnet7mpld0
```

Output:

```
{
  "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"
    ],
    "AllowedAuthFlowsUserPoolClient": false
}
}
```

- Per API i dettagli, vedere [DescribeUserPoolClient](#) in AWS CLI Command Reference.

## describe-user-pool-domain

Il seguente esempio di codice mostra come utilizzare `describe-user-pool-domain`.

### AWS CLI

Per descrivere un client per un pool di utenti

Questo esempio descrive un dominio del pool di utenti denominato `my-domain`.

Comando:

```
aws cognito-idp describe-user-pool-domain --domain my-domain
```

Output:

```
{
  "DomainDescription": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "AWSAccountId": "111111111111",
    "Domain": "my-domain",
    "S3Bucket": "aws-cognito-prod-pdx-assets",
    "CloudFrontDistribution": "aaaaaaaaaaaaa.cloudfront.net",
    "Version": "20190128175402",
    "Status": "ACTIVE",
    "CustomDomainConfig": {}
  }
}
```

```
}  
}
```

- Per API i dettagli, vedere [DescribeUserPoolDomain](#) in AWS CLI Command Reference.

## describe-user-pool

Il seguente esempio di codice mostra come utilizzare `describe-user-pool`.

### AWS CLI

Per descrivere un pool di utenti

Questo esempio descrive un pool di utenti con l'id del pool di utenti `us-west-2_aaaaaaaaa`.

Comando:

```
aws cognito-idp describe-user-pool --user-pool-id us-west-2_aaaaaaaaa
```

Output:

```
{  
  "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": {}
}
}
```

- Per i dettagli, vedere in Command Reference. API [DescribeUserPool](#)AWS CLI

## forget-device

Il seguente esempio di codice mostra come utilizzare `forget-device`.

### AWS CLI

Dimenticare un dispositivo

Questo esempio dimentica un dispositivo.

Comando:

```
aws cognito-idp forget-device --device-key us-west-2_abcd_1234-5678
```

- Per API i dettagli, vedere [ForgetDevice](#) in AWS CLI Command Reference.

## forgot-password

Il seguente esempio di codice mostra come utilizzare `forgot-password`.

## AWS CLI

Per forzare la modifica della password

L'forgot-passwordesempio seguente invia un messaggio a jane@example.com per modificare la password.

```
aws cognito-idp forgot-password --client-id 38fjsnc484p94kpbsnet7mpld0 --  
username jane@example.com
```

Output:

```
{  
  "CodeDeliveryDetails": {  
    "Destination": "j***@e***.com",  
    "DeliveryMedium": "EMAIL",  
    "AttributeName": "email"  
  }  
}
```

- Per API i dettagli, vedere [ForgotPassword](#) in AWS CLI Command Reference.

## get-csv-header

Il seguente esempio di codice mostra come utilizzare get-csv-header.

## AWS CLI

Per creare un'intestazione csv

Questo esempio crea un'intestazione csv.

Per ulteriori informazioni sull'importazione di utenti, vedere Importazione di utenti in pool di utenti da un file. CSV

Comando:

```
aws cognito-idp get-csv-header --user-pool-id us-west-2_aaaaaaaaa
```

Output:

```
{
  "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"
  ]
}
```

... Importazione di utenti in pool di utenti da un CSV file: <https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-using-import-tool.html>

- Per API i dettagli, vedere [GetCsvHeader](#) in AWS CLI Command Reference.

## get-group

Il seguente esempio di codice mostra come utilizzare `get-group`.

### AWS CLI

Per ottenere informazioni su un gruppo

Questo esempio ottiene informazioni su un gruppo denominato MyGroup.

Comando:

```
aws cognito-idp get-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup
```

Output:

```
{
  "Group": {
    "GroupName": "MyGroup",
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "Description": "A sample group.",
    "LastModifiedDate": 1548270073.795,
    "CreationDate": 1548270073.795
  }
}
```

- Per API i dettagli, vedere [GetGroup](#) in AWS CLI Command Reference.

## get-signing-certificate

Il seguente esempio di codice mostra come utilizzare `get-signing-certificate`.

AWS CLI

Per ottenere un certificato di firma

Questo esempio ottiene un certificato di firma per un pool di utenti.

Comando:

```
aws cognito-idp get-signing-certificate --user-pool-id us-west-2_aaaaaaaaa
```

Output:

```
{
  "Certificate": "CERTIFICATE_DATA"
}
```

- Per API i dettagli, vedere [GetSigningCertificate](#) in AWS CLI Command Reference.

## get-ui-customization

Il seguente esempio di codice mostra come utilizzare `get-ui-customization`.

## AWS CLI

Per ottenere informazioni sulla personalizzazione dell'interfaccia utente

Questo esempio ottiene informazioni sulla personalizzazione dell'interfaccia utente per un pool di utenti.

Comando:

```
aws cognito-idp get-ui-customization --user-pool-id us-west-2_aaaaaaaaa
```

Output:

```
{
  "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"
  }
}
```



```
}  
}
```

- Per API i dettagli, vedere [GetUiCustomization](#) in AWS CLI Command Reference.

## list-user-import-jobs

Il seguente esempio di codice mostra come utilizzare `list-user-import-jobs`.

### AWS CLI

Per elencare i lavori di importazione degli utenti

Questo esempio elenca i processi di importazione degli utenti.

Per ulteriori informazioni sull'importazione di utenti, vedere [Importazione di utenti in pool di utenti da un CSV file](#).

Comando:

```
aws cognito-idp list-user-import-jobs --user-pool-id us-west-2_aaaaaaaaaa --max-  
results 20
```

Output:

```
{  
  "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."
  }
]
}

```

- Per API i dettagli, vedere [ListUserImportJobs](#) in AWS CLI Command Reference.

## list-user-pools

Il seguente esempio di codice mostra come utilizzare `list-user-pools`.

### AWS CLI

Per elencare i pool di utenti

Questo esempio elenca fino a 20 pool di utenti.

Comando:

```
aws cognito-idp list-user-pools --max-results 20
```

Output:

```
{
  "UserPools": [
    {
      "CreationDate": 1547763720.822,
      "LastModifiedDate": 1547763720.822,
      "LambdaConfig": {},
      "Id": "us-west-2_aaaaaaaaa",
      "Name": "MyUserPool"
    }
  ]
}
```

- Per API i dettagli, vedere [ListUserPools](#) in AWS CLI Command Reference.

## list-users-in-group

Il seguente esempio di codice mostra come utilizzare `list-users-in-group`.

AWS CLI

Per elencare gli utenti in un gruppo

Questo esempio elenca gli utenti in un gruppo `MyGroup`.

Comando:

```
aws cognito-idp list-users-in-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup
```

Output:

```
{
  "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"
  }
]
}
```

- Per API i dettagli, vedere [ListUsersInGroup](#) in AWS CLI Command Reference.

## list-users

Il seguente esempio di codice mostra come utilizzare `list-users`.

### AWS CLI

Per elencare gli utenti

Questo esempio elenca fino a 20 utenti.

Comando:

```
aws cognito-idp list-users --user-pool-id us-west-2_aaaaaaaaa --limit 20
```

Output:

```
{
  "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"
        }
      ]
    }
  ]
}
```

- Per API i dettagli, vedere [ListUsers](#) in AWS CLI Command Reference.

## resend-confirmation-code

Il seguente esempio di codice mostra come utilizzare `resend-confirmation-code`.

### AWS CLI

Per inviare nuovamente un codice di conferma

L'esempio `resend-confirmation-code` seguente invia un codice di conferma all'utente `jane`.

```
aws cognito-idp resend-confirmation-code \  
  --client-id 12a3b456c7de890f11g123hijk \  
  --username jane
```

Output:

```
{  
  "CodeDeliveryDetails": {  
    "Destination": "j***@e***.com",  
    "DeliveryMedium": "EMAIL",  
    "AttributeName": "email"  
  }  
}
```

Per ulteriori informazioni, consulta [Registrazione e conferma degli account utente](#) nella Guida per gli sviluppatori di Amazon Cognito.

- Per API i dettagli, vedere [ResendConfirmationCode](#) in AWS CLI Command Reference.

## respond-to-auth-challenge

Il seguente esempio di codice mostra come utilizzare `respond-to-auth-challenge`.

### AWS CLI

Per rispondere a una richiesta di autenticazione

Questo esempio risponde a una richiesta di autorizzazione avviata con `initiate-auth`. È una risposta alla REQUIRED sfida NEW PASSWORD \_\_\_. Imposta una password per l'utente `jane@example.com`.

Comando:

```
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"
```

Output:

```
{
  "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"
    }
  }
}
```

- Per API i dettagli, vedere [RespondToAuthChallenge](#) in AWS CLI Command Reference.

## set-risk-configuration

Il seguente esempio di codice mostra come utilizzare `set-risk-configuration`.

AWS CLI

Per impostare la configurazione del rischio

Questo esempio imposta la configurazione del rischio per un pool di utenti. Imposta l'azione dell'evento di iscrizione su ACTION NO\_.

Comando:

```
aws cognito-idp set-risk-configuration --user-pool-id us-west-2_aaaaaaaaa --
compromised-credentials-risk-
configuration EventFilter=SIGN_UP,Actions={EventAction=NO_ACTION}
```

Output:

```
{
  "RiskConfiguration": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "CompromisedCredentialsRiskConfiguration": {
      "EventFilter": [
        "SIGN_UP"
      ],
      "Actions": {
        "EventAction": "NO_ACTION"
      }
    }
  }
}
```

- Per API i dettagli, vedere [SetRiskConfiguration](#) in AWS CLI Command Reference.

## set-ui-customization

Il seguente esempio di codice mostra come utilizzare `set-ui-customization`.

### AWS CLI

Per impostare la personalizzazione dell'interfaccia utente

Questo esempio personalizza l'CSS impostazione per un pool di utenti.

Comando:

```
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\n.inputField-  
customizable:focus {\n\tborder-color: #66afe9;\n\toutline: 0;\n}\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\n.idpButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:  
#31b0d5;\n}\n\n.socialButton-customizable {\n\theight: 40px;\n\ttext-align: left;  
\n\twidth: 100%;\n\tmargin-bottom: 15px;\n}\n\n.redirect-customizable {\n\ttext-  
align: center;\n}\n\n.passwordCheck-notValid-customizable {\n\tcolor: #DF3312;  
\n}\n\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;\n}\n\n.background-  
customizable {\n\tbackground-color: #faf;\n}\n\n"
```

## Output:

```
{
  "UICustomization": {
    "UserPoolId": "us-west-2_aaaaaaaaa",
    "ClientId": "ALL",
    "CSS": ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;  
\n}\n\n.banner-customizable {\n\tpadding: 25px 0px 25px 10px;\n\tbackground-color:  
lightgray;\n}\n\n.label-customizable {\n\tfont-weight: 300;\n}\n\n.textDescription-  
customizable {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;  
\n\tfont-size: 16px;\n}\n\n.idpDescription-customizable {\n\tpadding-top: 10px;\n  
\n\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n\n.legalText-  
customizable {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n\n.submitButton-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\n.submitButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:  
#286090;\n}\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\n.inputField-customizable {\n\twidth: 100%;\n\theight: 34px;\n\tcolor:  
#555;\n\tbackground-color: #fff;\n\tborder: 1px solid #ccc;\n}\n\n.inputField-  
customizable:focus {\n\tborder-color: #66afe9;\n\toutline: 0;\n}\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\n.idpButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:  
#31b0d5;\n}\n\n.socialButton-customizable {\n\theight: 40px;\n\ttext-align: left;  
\n\twidth: 100%;\n\tmargin-bottom: 15px;\n}\n\n.redirect-customizable {\n\ttext-  
align: center;\n}\n\n.passwordCheck-notValid-customizable {\n\tcolor: #DF3312;  
\n}\n\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;\n}\n\n.background-  
customizable {\n\tbackground-color: #faf;\n}\n\n",
    "CSSVersion": "20190129172214"
  }
}
```

- Per API i dettagli, vedere [SetUiCustomization](#) in AWS CLI Command Reference.

## set-user-mfa-preference

Il seguente esempio di codice mostra come utilizzare `set-user-mfa-preference`.

### AWS CLI

Per configurare MFA le impostazioni utente

L'esempio seguente modifica le opzioni MFA di consegna. Cambia il mezzo MFA di spedizione in 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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere MFA a un pool di utenti](#) nella Amazon Cognito Developer Guide.

- Per API i dettagli, consulta [SetUserMfaPreference AWS CLI Command Reference](#).

## set-user-settings

Il seguente esempio di codice mostra come utilizzare `set-user-settings`.

### AWS CLI

Per configurare le impostazioni utente

Questo esempio imposta la preferenza MFA di consegna su EMAIL.

Comando:

```
aws cognito-idp set-user-settings --access-token ACCESS_TOKEN --mfa-
options DeliveryMedium=EMAIL
```

- Per API i dettagli, vedere [SetUserSettings](#) in AWS CLI Command Reference.

## sign-up

Il seguente esempio di codice mostra come utilizzare `sign-up`.

### AWS CLI

Per registrare un utente

In questo esempio viene registrato `jane@example.com`.

Comando:

```
aws cognito-idp sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --  
username jane@example.com --password PASSWORD --user-attributes  
Name="email",Value="jane@example.com" Name="name",Value="Jane"
```

Output:

```
{  
  "UserConfirmed": false,  
  "UserSub": "e04d60a6-45dc-441c-a40b-e25a787d4862"  
}
```

- Per API i dettagli, vedere [SignUp](#) in AWS CLI Command Reference.

## start-user-import-job

Il seguente esempio di codice mostra come utilizzare `start-user-import-job`.

### AWS CLI

Per avviare un processo di importazione degli utenti

Questo esempio avvia un processo di input da parte dell'utente.

Per ulteriori informazioni sull'importazione di utenti, vedere [Importazione di utenti in pool di utenti da un CSV file](#).

Comando:

```
aws cognito-idp start-user-import-job --user-pool-id us-west-2_aaaaaaaaa --job-  
id import-TZqNQvDRnW
```

**Output:**

```
{
  "UserImportJob": {
    "JobName": "import-Test10",
    "JobId": "import-lmpxS0uIzH",
    "UserPoolId": "us-west-2_aaaaaaaaaa",
    "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
  }
}
```

- Per API i dettagli, vedere [StartUserImportJob](#) in AWS CLI Command Reference.

**stop-user-import-job**

Il seguente esempio di codice mostra come utilizzare `stop-user-import-job`.

**AWS CLI**

Per interrompere un processo di importazione utente

Questo esempio interrompe un processo di input da parte dell'utente.

Per ulteriori informazioni sull'importazione di utenti, vedere [Importazione di utenti in pool di utenti da un CSV file](#).

Comando:

```
aws cognito-idp stop-user-import-job --user-pool-id us-west-2_aaaaaaaaaa --job-
id import-TZqNQvDRnW
```

**Output:**

```
{
```

```

  "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."
  }
}

```

- Per API i dettagli, vedere [StopUserImportJob](#) in AWS CLI Command Reference.

## update-auth-event-feedback

Il seguente esempio di codice mostra come utilizzare `update-auth-event-feedback`.

### AWS CLI

Per aggiornare il feedback sugli eventi di autenticazione

Questo esempio aggiorna il feedback sugli eventi di autorizzazione. Contrassegna l'evento come «Valido».

Comando:

```

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"

```

- Per API i dettagli, vedere [UpdateAuthEventFeedback](#) in AWS CLI Command Reference.

## update-device-status

Il seguente esempio di codice mostra come utilizzare `update-device-status`.

## AWS CLI

Per aggiornare lo stato del dispositivo

Questo esempio aggiorna lo stato di un dispositivo a «not\_memored».

Comando:

```
aws cognito-idp update-device-status --access-token ACCESS_TOKEN --device-key DEVICE_KEY --device-remembered-status "not_remembered"
```

- Per API i dettagli, vedere [UpdateDeviceStatus](#) in AWS CLI Command Reference.

## update-group

Il seguente esempio di codice mostra come utilizzare `update-group`.

### AWS CLI

Per aggiornare un gruppo

Questo esempio aggiorna la descrizione e la precedenza di MyGroup.

Comando:

```
aws cognito-idp update-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup --description "New description" --precedence 2
```

Output:

```
{
  "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
  }
}
```

- Per API i dettagli, vedere [UpdateGroup](#) in AWS CLI Command Reference.

## update-resource-server

Il seguente esempio di codice mostra come utilizzare `update-resource-server`.

### AWS CLI

Per aggiornare un server di risorse

Questo esempio aggiorna il server di risorse Weather. Aggiunge un nuovo ambito.

Comando:

```
aws cognito-idp update-resource-server --user-pool-id us-west-2_aaaaaaaa
--identifier weather.example.com --name Weather --scopes
ScopeName=NewScope,ScopeDescription="New scope description"
```

Output:

```
{
  "ResourceServer": {
    "UserPoolId": "us-west-2_aaaaaaaa",
    "Identifier": "weather.example.com",
    "Name": "Happy",
    "Scopes": [
      {
        "ScopeName": "NewScope",
        "ScopeDescription": "New scope description"
      }
    ]
  }
}
```

- Per API i dettagli, vedere [UpdateResourceServer](#) in AWS CLI Command Reference.

## update-user-attributes

Il seguente esempio di codice mostra come utilizzare `update-user-attributes`.

### AWS CLI

Per aggiornare gli attributi utente

Questo esempio aggiorna l'attributo utente «nickname».

Comando:

```
aws cognito-idp update-user-attributes --access-token ACCESS_TOKEN --user-attributes  
Name="nickname",Value="Dan"
```

- Per API i dettagli, vedere [UpdateUserAttributes](#) in AWS CLI Command Reference.

## update-user-pool-client

Il seguente esempio di codice mostra come utilizzare `update-user-pool-client`.

AWS CLI

Per aggiornare un client del pool di utenti

Questo esempio aggiorna il nome di un client del pool di utenti. Aggiunge anche un attributo scrivibile «nickname».

Comando:

```
aws cognito-idp update-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --  
client-id 3n4b5urk1ft4fl3mg5e62d9ado --client-name "NewClientName" --write-  
attributes "nickname"
```

Output:

```
{  
  "UserPoolClient": {  
    "UserPoolId": "us-west-2_aaaaaaaaa",  
    "ClientName": "NewClientName",  
    "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",  
    "LastModifiedDate": 1548802761.334,  
    "CreationDate": 1548178931.258,  
    "RefreshTokenValidity": 30,  
    "WriteAttributes": [  
      "nickname"  
    ],  
    "AllowedOAuthFlowsUserPoolClient": false  
  }  
}
```



- Per API i dettagli, vedere [UpdateUserPoolClient](#) in AWS CLI Command Reference.

## update-user-pool

Il seguente esempio di codice mostra come utilizzare `update-user-pool`.

### AWS CLI

Per aggiornare un pool di utenti

L'update-user-pool esempio seguente modifica un pool di utenti con una sintassi di esempio per ciascuna delle opzioni di configurazione disponibili. Per aggiornare un pool di utenti, è necessario specificare tutte le opzioni configurate in precedenza o verranno ripristinate al valore predefinito.

```
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=
\[\{Priority=1,Name="verified_email"\},\{Priority=2,Name="verified_phone_number"\}\]

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiornamento della configurazione del pool di utenti](#) nella Amazon Cognito Developer Guide.

- Per API i dettagli, consulta [UpdateUserPool AWS CLI Command Reference](#).

## Esempi di Amazon Comprehend con AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Comprehend. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **batch-detect-dominant-language**

Il seguente esempio di codice mostra come utilizzare `batch-detect-dominant-language`.

## AWS CLI

Per rilevare la lingua dominante di più testi in ingresso

L'batch-detect-dominant-languageesempio seguente analizza più testi di input e restituisce la lingua dominante di ciascuno di essi. Per ogni previsione viene inoltre emesso il punteggio di confidenza dei modelli preaddestrati.

```
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."
```

Output:

```
{  
  "ResultList": [  
    {  
      "Index": 0,  
      "Languages": [  
        {  
          "LanguageCode": "en",  
          "Score": 0.9986501932144165  
        }  
      ]  
    }  
  ],  
  "ErrorList": []  
}
```

Per ulteriori informazioni, consulta [Dominant Language](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [BatchDetectDominantLanguage AWS CLI Command Reference](#).

## batch-detect-entities

Il seguente esempio di codice mostra come utilizzarebatch-detect-entities.

## AWS CLI

Per rilevare entità da più testi di input

L'batch-detect-entitiesesempio seguente analizza più testi di input e restituisce le entità denominate di ciascuno di essi. Per ogni previsione viene inoltre emesso il punteggio di confidenza del modello pre-addestrato.

```
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."
```

Output:

```
{  
  "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": []
}

```

Per ulteriori informazioni, consulta [Entities](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [BatchDetectEntities AWS CLI Command Reference](#).

## batch-detect-key-phrases

Il seguente esempio di codice mostra come utilizzare `batch-detect-key-phrases`.

### AWS CLI

Per rilevare frasi chiave di più input di testo

L'`batch-detect-key-phrases` esempio seguente analizza più testi di input e restituisce le frasi nominali chiave di ciascuno di essi. Viene inoltre emesso il punteggio di confidenza del modello pre-addestrato per ogni previsione.

```

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."

```

Output:

```

{
  "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": []
}

```

Per ulteriori informazioni, consulta le [frasi chiave](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [BatchDetectKeyPhrasesReference](#).

## batch-detect-sentiment

Il seguente esempio di codice mostra come utilizzare `batch-detect-sentiment`.

### AWS CLI

Per rilevare il sentimento prevalente di più testi in input

L'`batch-detect-sentiment` seguente analizza più testi di input e restituisce il sentimento prevalente (POSITIVE,, o NEUTRALMIXED, di ciascuno di essi). NEGATIVE

```

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

```

Output:

```
{
```

```
"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": []  
}
```

Per ulteriori informazioni, consulta [Sentiment](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [BatchDetectSentiment](#)Reference.

## batch-detect-syntax

Il seguente esempio di codice mostra come utilizzare `batch-detect-syntax`.

## AWS CLI

Per esaminare la sintassi e le parti del discorso delle parole in più testi di input

L'batch-detect-syntaxesempio seguente analizza la sintassi di più testi di input e restituisce le diverse parti del discorso. Per ogni previsione viene inoltre emesso il punteggio di confidenza del modello pre-addestrato.

```
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
```

Output:

```
{  
  "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": []
}

```

Per ulteriori informazioni, consulta [Syntax Analysis](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [BatchDetectSyntax](#) Reference.

## batch-detect-targeted-sentiment

Il seguente esempio di codice mostra come utilizzare `batch-detect-targeted-sentiment`.

### AWS CLI

Per rilevare il sentimento e ogni entità denominata per più testi di input

L'esempio seguente analizza più testi di input e restituisce le entità denominate insieme al sentimento prevalente associato a ciascuna entità. Per ogni previsione viene inoltre emesso il punteggio di confidenza del modello pre-addestrato.



```
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."
```

Output:

```
{  
  "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": []
}

```

Per ulteriori informazioni, consulta [Targeted Sentiment](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [BatchDetectTargetedSentiment](#) Reference.

## classify-document

Il seguente esempio di codice mostra come utilizzare `classify-document`.

### AWS CLI

Per classificare un documento con un endpoint specifico del modello

L'esempio seguente classifica un documento con un punto finale di un modello personalizzato. Il modello in questo esempio è stato addestrato su un set di dati contenente messaggi sms etichettati come spam o non spam o «ham».

```
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"
```

Output:

```
{  
  "Classes": [  
    {  
      "Name": "spam",  
      "Score": 0.9998599290847778  
    },  
    {  
      "Name": "ham",  
      "Score": 0.00014001205272506922  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta la sezione [Classificazione personalizzata](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [ClassifyDocument AWS CLI](#) Command Reference.

## contains-pii-entities

Il seguente esempio di codice mostra come utilizzare `contains-pii-entities`.

## AWS CLI

Per analizzare il testo di input per verificare la presenza di PII informazioni

L'contains-pii-entities esempio seguente analizza il testo di input per verificare la presenza di informazioni di identificazione personale (PII) e restituisce le etichette dei tipi di PII entità identificati, quali nome, indirizzo, numero di conto corrente bancario o numero di telefono.

```
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."
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

```
}
```

Per ulteriori informazioni, consulta [Personally Identifiable Information \(PII\)](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ContainsPiiEntities](#) Reference.

## create-dataset

Il seguente esempio di codice mostra come utilizzare `create-dataset`.

### AWS CLI

Per creare un set di dati Flywheel

L'`create-dataset` esempio seguente crea un set di dati per un volano. Questo set di dati verrà utilizzato come dati di addestramento aggiuntivi, come specificato dal tag. `--dataset-type`

```
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
```

Contenuto di `file://inputConfig.json`.

```
{  
  "DataFormat": "COMPREHEND_CSV",  
  "DocumentClassifierInputDataConfig": {  
    "S3Uri": "s3://DOC-EXAMPLE-BUCKET/training-data.csv"  
  }  
}
```

Output:

```
{  
  "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity/dataset/example-dataset"  
}
```

Per ulteriori informazioni, consulta [Flywheel Overview](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateDatasetReference](#).

## create-document-classifier

Il seguente esempio di codice mostra come utilizzare `create-document-classifier`.

### AWS CLI

Per creare un classificatore di documenti per classificare i documenti

L'esempio seguente inizia il processo di formazione per un modello di classificatore di documenti. Il file dei dati di addestramento `training.csv`, si trova nel `--input-data-config` tag. `training.csv` è un documento a due colonne in cui le etichette o le classificazioni sono fornite nella prima colonna e i documenti sono forniti nella seconda colonna.

```
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
```

Output:

```
{
  "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier"
}
```

Per ulteriori informazioni, consulta la sezione [Classificazione personalizzata](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [CreateDocumentClassifier AWS CLI Command Reference](#).

## create-endpoint

Il seguente esempio di codice mostra come utilizzare `create-endpoint`.



## AWS CLI

Per creare un endpoint per un modello personalizzato

L'`create-endpoint` esempio seguente crea un endpoint per l'inferenza sincrona per un modello personalizzato precedentemente addestrato.

```
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
```

Output:

```
{  
  "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint-1"  
}
```

Per ulteriori informazioni, consulta [Managing Amazon Comprehend endpoint nella Amazon Comprehend Developer Guide](#).

- Per API i dettagli, consulta Command [CreateEndpoint](#) Reference AWS CLI .

## `create-entity-recognizer`

Il seguente esempio di codice mostra come utilizzare `create-entity-recognizer`.

## AWS CLI

Per creare un riconoscitore di entità personalizzato

L'`create-entity-recognizer` esempio seguente inizia il processo di formazione per un modello di riconoscimento di entità personalizzato. Questo esempio utilizza un CSV file contenente documenti di addestramento e un elenco di CSV entità `entity_list.csv` per addestrare il modello. `raw_text.csv` `entity-list.csv` contiene le seguenti colonne: testo e tipo.

```
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

```

Output:

```

{
  "EntityRecognizerArn": "arn:aws:comprehend:us-west-2:111122223333:example-
entity-recognizer/entityrecognizer1"
}

```

Per ulteriori informazioni, consulta il [riconoscimento personalizzato delle entità](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [CreateEntityRecognizer AWS CLI Command Reference](#).

## create-flywheel

Il seguente esempio di codice mostra come utilizzare `create-flywheel`.

### AWS CLI

Per creare un volano

L'`create-flywheel` esempio seguente crea un volano per orchestrare la formazione continua di un modello di classificazione dei documenti o di riconoscimento delle entità. Il volano in questo esempio viene creato per gestire un modello addestrato esistente specificato dal tag. `--active-model-arn` Quando viene creato il volano, viene creato un data lake sul tag. `--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"

```

Output:

```
{
  "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel"
}
```

Per ulteriori informazioni, consulta [Flywheel Overview](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateFlywheel](#)Reference.

## **delete-document-classifier**

Il seguente esempio di codice mostra come utilizzare `delete-document-classifier`.

### AWS CLI

Per eliminare un classificatore di documenti personalizzato

L'`delete-document-classifier` esempio seguente elimina un modello di classificatore di documenti personalizzato.

```
aws comprehend delete-document-classifier \
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier-1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing Amazon Comprehend endpoint nella Amazon Comprehend](#) Developer Guide.

- Per API i dettagli, consulta Command [DeleteDocumentClassifier](#)Reference AWS CLI .

## **delete-endpoint**

Il seguente esempio di codice mostra come utilizzare `delete-endpoint`.

### AWS CLI

Per eliminare un endpoint per un modello personalizzato

L'`delete-endpoint` esempio seguente elimina un endpoint specifico del modello. Per eliminare il modello, è necessario eliminare tutti gli endpoint.

```
aws comprehend delete-endpoint \  
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint-1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing Amazon Comprehend endpoint nella Amazon Comprehend Developer Guide](#).

- Per API i dettagli, consulta Command [DeleteEndpoint](#)Reference AWS CLI .

## delete-entity-recognizer

Il seguente esempio di codice mostra come utilizzare `delete-entity-recognizer`.

AWS CLI

Per eliminare un modello di riconoscimento delle entità personalizzato

L'`delete-entity-recognizer` esempio seguente elimina un modello di riconoscimento delle entità personalizzato.

```
aws comprehend delete-entity-recognizer \  
  --entity-recognizer-arn arn:aws:comprehend:us-west-2:111122223333:entity-  
recognizer/example-entity-recognizer-1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing Amazon Comprehend endpoint nella Amazon Comprehend Developer Guide](#).

- Per API i dettagli, consulta Command [DeleteEntityRecognizer](#)Reference AWS CLI .

## delete-flywheel

Il seguente esempio di codice mostra come utilizzare `delete-flywheel`.

AWS CLI

Per eliminare un volano

L'`delete-flywheel` esempio seguente elimina un volano. Il data lake o il modello associato al volano non vengono eliminati.

```
aws comprehend delete-flywheel \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-  
flywheel-1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la [panoramica di Flywheel](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteFlywheel](#) Reference.

## `delete-resource-policy`

Il seguente esempio di codice mostra come utilizzare `delete-resource-policy`.

### AWS CLI

Per eliminare una politica basata sulle risorse

L'`delete-resource-policy` esempio seguente elimina una policy basata sulle risorse da una risorsa Amazon Comprehend.

```
aws comprehend delete-resource-policy \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier-1/version/1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [consulta Copiare modelli personalizzati tra AWS account](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteResourcePolicy](#) Reference.

## `describe-dataset`

Il seguente esempio di codice mostra come utilizzare `describe-dataset`.

## AWS CLI

Per descrivere un set di dati Flywheel

L'`describe-dataset`esempio seguente ottiene le proprietà di un set di dati Flywheel.

```
aws comprehend describe-dataset \  
  --dataset-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity/dataset/example-dataset
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Flywheel Overview](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeDataset](#)Reference.

## describe-document-classification-job

Il seguente esempio di codice mostra come utilizzare `describe-document-classification-job`.

### AWS CLI

Descrivere un lavoro di classificazione dei documenti

L'`describe-document-classification-job`esempio seguente ottiene le proprietà di un processo asincrono di classificazione dei documenti.

```
aws comprehend describe-document-classification-job \  
  --job-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity/job/example-job
```

```
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta la sezione [Classificazione personalizzata](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [DescribeDocumentClassificationJob AWS CLI Command Reference](#).

## describe-document-classifier

Il seguente esempio di codice mostra come utilizzare `describe-document-classifier`.

### AWS CLI

Per descrivere un classificatore di documenti

L'esempio seguente ottiene le proprietà di un modello di classificatore di documenti personalizzato.

```
aws comprehend describe-document-classifier \  
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-  
classifier/example-classifier-1
```

Output:

```
{  
  "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"  
  }  
}
```



Per ulteriori informazioni, consulta [Creazione e gestione di modelli personalizzati](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [DescribeDocumentClassifier AWS CLI Command Reference](#).

## describe-dominant-language-detection-job

Il seguente esempio di codice mostra come utilizzare `describe-dominant-language-detection-job`.

### AWS CLI

Descrivere un lavoro di rilevamento della lingua dominante.

L'esempio seguente ottiene le proprietà di un processo asincrono di rilevamento della lingua dominante.

```
aws comprehend describe-dominant-language-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [DescribeDominantLanguageDetectionJobCommand](#) Reference.AWS CLI

## describe-endpoint

Il seguente esempio di codice mostra come utilizzare `describe-endpoint`.

### AWS CLI

Per descrivere un endpoint specifico

L'`describe-endpoint`esempio seguente ottiene le proprietà di un endpoint specifico del modello.

```
aws comprehend describe-endpoint \  
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
  endpoint/example-classifier-endpoint
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Managing Amazon Comprehend endpoint nella Amazon Comprehend Developer Guide](#).

- Per API i dettagli, consulta Command [DescribeEndpoint](#)Reference AWS CLI .

## describe-entities-detection-job

Il seguente esempio di codice mostra come utilizzare `describe-entities-detection-job`.

### AWS CLI

Per descrivere un processo di rilevamento delle entità

L'esempio seguente ottiene le proprietà di un processo di rilevamento di entità asincrono.

```
aws comprehend describe-entities-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [DescribeEntitiesDetectionJob](#) Command Reference.AWS CLI

## describe-entity-recognizer

Il seguente esempio di codice mostra come utilizzare `describe-entity-recognizer`.

### AWS CLI

Per descrivere un identificatore di entità

L'esempio seguente ottiene le proprietà di un modello di riconoscimento di entità personalizzato.

```
aws comprehend describe-entity-recognizer \
    entity-recognizer-arn arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/business-recongizer-1/version/1
```

Output:

```
{
  "EntityRecognizerProperties": {
    "EntityRecognizerArn": "arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/business-recongizer-1/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-example-role",
  "VersionName": "1"
}
}

```

Per ulteriori informazioni, consulta il [riconoscimento personalizzato delle entità](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [DescribeEntityRecognizer AWS CLI Command Reference](#).

## describe-events-detection-job

Il seguente esempio di codice mostra come utilizzare `describe-events-detection-job`.

### AWS CLI

Per descrivere un processo di rilevamento degli eventi.

L'esempio seguente ottiene le proprietà di un processo asincrono di rilevamento degli eventi.

```

aws comprehend describe-events-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE

```

**Output:**

```
{
  "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"
    ]
  }
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [DescribeEventsDetectionJob](#) Command Reference.AWS CLI

**describe-flywheel-iteration**

Il seguente esempio di codice mostra come utilizzare `describe-flywheel-iteration`.

**AWS CLI**

Per descrivere un'iterazione del volano

L'`describe-flywheel-iteration` seguente ottiene le proprietà di un'iterazione del volano.

```
aws comprehend describe-flywheel-iteration \
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-  
flywheel \
  --flywheel-iteration-id 20232222AEXAMPLE
```

Output:

```
{
  "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/"
  }
}
```

Per ulteriori informazioni, consulta la [panoramica di Flywheel](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeFlywheelIteration](#) Reference.

## describe-flywheel

Il seguente esempio di codice mostra come utilizzare `describe-flywheel`.

### AWS CLI

Descrivere un volano

L'`describe-flywheel` esempio seguente ottiene le proprietà di un volano. In questo esempio, il modello associato al volano è un modello di classificatore personalizzato addestrato a classificare i documenti come spam o non spam oppure come «ham».

```
aws comprehend describe-flywheel \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-  
flywheel
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Flywheel Overview](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeFlywheelReference](#).

## describe-key-phrases-detection-job

Il seguente esempio di codice mostra come utilizzare `describe-key-phrases-detection-job`.

### AWS CLI

Per descrivere un processo di rilevamento di frasi chiave

L'`describe-key-phrases-detection-job`esempio seguente ottiene le proprietà di un processo di rilevamento asincrono di frasi chiave.

```

aws comprehend describe-key-phrases-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [DescribeKeyPhrasesDetectionJob](#) Command Reference.AWS CLI

## describe-pii-entities-detection-job

Il seguente esempio di codice mostra come utilizzare `describe-pii-entities-detection-job`.

### AWS CLI

Per descrivere un processo di rilevamento delle PII entità

L'`describe-pii-entities-detection-job` esempio seguente ottiene le proprietà di un processo asincrono di rilevamento di entità pii.

```

aws comprehend describe-pii-entities-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE

```

Output:

```

{
  "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"
    }
}

```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [DescribePiiEntitiesDetectionJob](#) Command Reference.AWS CLI

## describe-resource-policy

Il seguente esempio di codice mostra come utilizzare `describe-resource-policy`.

### AWS CLI

Per descrivere una politica delle risorse allegata a un modello

L'`describe-resource-policy`esempio seguente ottiene le proprietà di una politica basata sulle risorse allegata a un modello.

```

aws comprehend describe-resource-policy \
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1

```

Output:

```

{
  "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"
}

```

Per ulteriori informazioni, [consulta Copiare modelli personalizzati tra AWS account](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeResourcePolicy](#) Reference.

## describe-sentiment-detection-job

Il seguente esempio di codice mostra come utilizzare `describe-sentiment-detection-job`.

### AWS CLI

Descrivere un lavoro di rilevamento dei sentimenti

L'`describe-sentiment-detection-job`esempio seguente ottiene le proprietà di un processo asincrono di rilevamento del sentiment.

```
aws comprehend describe-sentiment-detection-job \  
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [DescribeSentimentDetectionJob](#) Command Reference.AWS CLI

## describe-targeted-sentiment-detection-job

Il seguente esempio di codice mostra come utilizzare `describe-targeted-sentiment-detection-job`.

### AWS CLI

Per descrivere un lavoro mirato di rilevamento dei sentimenti

L'esempio seguente ottiene le proprietà di un processo asincrono mirato di rilevamento del sentiment.

```
aws comprehend describe-targeted-sentiment-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Output:

```
{
  "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"
  }
}
```

```
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [DescribeTargetedSentimentDetectionJob](#) Command Reference.AWS CLI

## describe-topics-detection-job

Il seguente esempio di codice mostra come utilizzare `describe-topics-detection-job`.

### AWS CLI

Per descrivere un lavoro di rilevamento di argomenti

L'`describe-topics-detection-job`esempio seguente ottiene le proprietà di un processo asincrono di rilevamento degli argomenti.

```
aws comprehend describe-topics-detection-job \  
--job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Output:

```
{  
  "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"  
  }  
}
```

```
}  
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [DescribeTopicsDetectionJob](#) Command Reference.AWS CLI

## detect-dominant-language

Il seguente esempio di codice mostra come utilizzare `detect-dominant-language`.

### AWS CLI

Per rilevare la lingua dominante del testo di input

Quanto segue `detect-dominant-language` analizza il testo di input e identifica la lingua dominante. Viene inoltre emesso il punteggio di confidenza del modello pre-addestrato.

```
aws comprehend detect-dominant-language \  
  --text "It is a beautiful day in Seattle."
```

Output:

```
{  
  "Languages": [  
    {  
      "LanguageCode": "en",  
      "Score": 0.9877256155014038  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Dominant Language](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [DetectDominantLanguage AWS CLI](#) Command Reference.

## detect-entities

Il seguente esempio di codice mostra come utilizzare `detect-entities`.

## AWS CLI

Per rilevare entità denominate nel testo di input

L'entità `detect-entities` seguente analizza il testo di input e restituisce le entità denominate. Per ogni previsione viene inoltre emesso il punteggio di confidenza del modello pre-addestrato.

```
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."
```

Output:

```
{  
  "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
  }
]
}

```

Per ulteriori informazioni, consulta [Entities](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [DetectEntities AWS CLI Command Reference](#).

## detect-key-phrases

Il seguente esempio di codice mostra come utilizzare `detect-key-phrases`.

### AWS CLI

Per rilevare le frasi chiave nel testo di input

L'esempio seguente analizza il testo di input e identifica le frasi nominali chiave. Per ogni previsione viene inoltre emesso il punteggio di confidenza del modello pre-addestrato.

```

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."***

**Output:**

```
{
  "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
  }
]
}
```

Per ulteriori informazioni, consulta le [frasi chiave](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DetectKeyPhrasesReference](#).

## **detect-pii-entities**

Il seguente esempio di codice mostra come utilizzare `detect-pii-entities`.

### AWS CLI

Per rilevare le entità pii nel testo di input

L' `detect-pii-entities` seguente analizza il testo di input e identifica le entità che contengono informazioni di identificazione personale (PII). Per ogni previsione viene inoltre emesso il punteggio di confidenza del modello pre-addestrato.

```
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."
```

Output:

```
{
  "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
}
]
```

Per ulteriori informazioni, consulta [Personally Identifiable Information \(PII\)](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DetectPiiEntitiesReference](#).

## **detect-sentiment**

Il seguente esempio di codice mostra come utilizzare `detect-sentiment`.

## AWS CLI

Per rilevare il sentimento di un testo di input

L'`detect-sentiment` seguente analizza il testo di input e restituisce un'inferenza del sentimento prevalente (POSITIVE, o). NEUTRAL MIXED NEGATIVE

```
aws comprehend detect-sentiment \  
  --language-code en \  
  --text "It is a beautiful day in Seattle"
```

Output:

```
{  
  "Sentiment": "POSITIVE",  
  "SentimentScore": {  
    "Positive": 0.9976957440376282,  
    "Negative": 9.653854067437351e-05,  
    "Neutral": 0.002169104292988777,  
    "Mixed": 3.857641786453314e-05  
  }  
}
```

Per ulteriori informazioni, consulta [Sentiment](#) nella Amazon Comprehend Developer Guide

- Per API i dettagli, consulta AWS CLI Command [DetectSentiment](#) Reference.

## `detect-syntax`

Il seguente esempio di codice mostra come utilizzare `detect-syntax`.

## AWS CLI

Per rilevare le parti del parlato in un testo di input

L'`detect-syntax` seguente analizza la sintassi del testo di input e restituisce le diverse parti del discorso. Per ogni previsione viene inoltre emesso il punteggio di confidenza del modello pre-addestrato.

```
aws comprehend detect-syntax \  
  --language-code en \  
  --text "It is a beautiful day in Seattle."
```



## Output:

```
{
  "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
  }
}
]
```

Per ulteriori informazioni, consulta [Syntax Analysis](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DetectSyntax](#)Reference.

## **detect-targeted-sentiment**

Il seguente esempio di codice mostra come utilizzare `detect-targeted-sentiment`.

### AWS CLI

Per rilevare il sentimento mirato delle entità denominate in un testo di input

L'entità `detect-targeted-sentiment` seguente analizza il testo di input e restituisce le entità denominate oltre al sentimento mirato associato a ciascuna entità. Viene inoltre emesso il punteggio di confidenza dei modelli preaddestrati per ogni previsione.

```
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"
```

Output:

```
{  
  "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": 3.999999989900971e-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
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Targeted Sentiment](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DetectTargetedSentiment](#)Reference.

## import-model

Il seguente esempio di codice mostra come utilizzare `import-model`.

### AWS CLI

Per importare un modello

L'esempio seguente importa un modello da un AWS account diverso. Il modello di classificazione dei documenti in account 444455556666 ha una politica basata sulle risorse che consente 111122223333 all'account di importare il modello.

```
aws comprehend import-model \  
  --source-model-arn arn:aws:comprehend:us-west-2:444455556666:document-  
classifier/example-classifier
```

Output:

```
{  
  "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier"  
}
```

Per ulteriori informazioni, [consulta Copiare modelli personalizzati tra AWS account](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ImportModel](#)Reference.

## list-datasets

Il seguente esempio di codice mostra come utilizzare `list-datasets`.

### AWS CLI

Per elencare tutti i set di dati Flywheel

L'`list-datasets` esempio seguente elenca tutti i set di dati associati a un volano.

```
aws comprehend list-datasets \  
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-  
entity
```

Output:

```
{  
  "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"
    }
]
}
```

Per ulteriori informazioni, consulta [Flywheel Overview](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListDatasets](#) Reference.

## list-document-classification-jobs

Il seguente esempio di codice mostra come utilizzare `list-document-classification-jobs`.

### AWS CLI

Per un elenco di tutti i lavori di classificazione dei documenti

L'`list-document-classification-jobs` esempio seguente elenca tutti i processi di classificazione dei documenti.

```
aws comprehend list-document-classification-jobs
```

Output:

```
{
  "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": "123456abcdeb0e11022f22a1EXAMPLE2",
    "JobArn": "arn:aws:comprehend:us-west-2:1234567890101:document-
classification-job/123456abcdeb0e11022f22a1EXAMPLE2",
    "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"
  }
]
}

```

Per ulteriori informazioni, consulta la sezione [Classificazione personalizzata](#) nella Amazon Comprehend Developer Guide.



- Per API i dettagli, consulta [ListDocumentClassificationJobs AWS CLI Command Reference](#).

## **list-document-classifier-summaries**

Il seguente esempio di codice mostra come utilizzare `list-document-classifier-summaries`.

### AWS CLI

Per elencare i riepiloghi di tutti i classificatori di documenti creati

L'`list-document-classifier-summaries` esempio seguente elenca tutti i riepiloghi dei classificatori di documenti creati.

```
aws comprehend list-document-classifier-summaries
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Creazione e gestione di modelli personalizzati](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [ListDocumentClassifierSummaries AWS CLI Command Reference](#).

## list-document-classifiers

Il seguente esempio di codice mostra come utilizzare `list-document-classifiers`.

### AWS CLI

Per un elenco di tutti i classificatori di documenti

L'`list-document-classifiers` seguente elenca tutti i modelli di classificatori di documenti addestrati e in corso di formazione.

```
aws comprehend list-document-classifiers
```

Output:

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Creazione e gestione di modelli personalizzati](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [ListDocumentClassifiers AWS CLI Command Reference](#).

## list-dominant-language-detection-jobs

Il seguente esempio di codice mostra come utilizzare `list-dominant-language-detection-jobs`.

### AWS CLI

Per elencare tutti i lavori di rilevamento della lingua dominante

L'esempio seguente elenca tutti i processi di rilevamento della lingua dominante asincroni in corso e completati.

**aws comprehend list-dominant-language-detection-jobs**

## Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [ListDominantLanguageDetectionJobs](#) Command Reference.AWS CLI

## list-endpoints

Il seguente esempio di codice mostra come utilizzare `list-endpoints`.

AWS CLI

Per elencare tutti gli endpoint

L'`list-endpoints` esempio seguente elenca tutti gli endpoint attivi specifici del modello.

```
aws comprehend list-endpoints
```

Output:

```

{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Managing Amazon Comprehend endpoint nella Amazon Comprehend Developer Guide](#).

- Per API i dettagli, consulta Command [ListEndpoints](#)Reference AWS CLI .

## list-entities-detection-jobs

Il seguente esempio di codice mostra come utilizzare `list-entities-detection-jobs`.

### AWS CLI

Per elencare tutti i processi di rilevamento delle entità

L'`list-entities-detection-jobs`esempio seguente elenca tutti i processi asincroni di rilevamento delle entità.

```
aws comprehend list-entities-detection-jobs
```

Output:

```

{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Entities](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [ListEntitiesDetectionJobs AWS CLI Command Reference](#).

## list-entity-recognizer-summaries

Il seguente esempio di codice mostra come utilizzare `list-entity-recognizer-summaries`.

### AWS CLI

All'elenco dei riepiloghi di tutti i riconoscitori di entità creati

L'`list-entity-recognizer-summaries` seguente elenca tutti i riepiloghi di Entity Recognizer.

```
aws comprehend list-entity-recognizer-summaries
```

Output:

```

{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta il [riconoscimento personalizzato delle entità](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [ListEntityRecognizerSummaries AWS CLI Command Reference](#).

## list-entity-recognizers

Il seguente esempio di codice mostra come utilizzare `list-entity-recognizers`.

### AWS CLI

Per un elenco di tutti i riconoscitori di entità personalizzati

L'`list-entity-recognizers` seguente elenca tutti i riconoscitori di entità personalizzati creati.

```
aws comprehend list-entity-recognizers
```

Output:

```

{
  "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"
]
```

```
}
```

Per ulteriori informazioni, consulta il [riconoscimento personalizzato delle entità](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [ListEntityRecognizers AWS CLI Command Reference](#).

## list-events-detection-jobs

Il seguente esempio di codice mostra come utilizzare `list-events-detection-jobs`.

### AWS CLI

Per elencare tutti i processi di rilevamento degli eventi

L'`list-events-detection-jobs` seguente elenca tutti i processi asincroni di rilevamento degli eventi.

```
aws comprehend list-events-detection-jobs
```

Output:

```
{
  "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/111122223333-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"
        ]
    }
]
}
}

```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [ListEventsDetectionJobs](#) Command Reference.AWS CLI

## list-flywheel-iteration-history

Il seguente esempio di codice mostra come utilizzare `list-flywheel-iteration-history`.

### AWS CLI

Per elencare tutta la cronologia delle iterazioni del volano

L'esempio seguente elenca tutte le iterazioni di un volano.

```
aws comprehend list-flywheel-iteration-history
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel
```

Output:

```
{
  "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,
        "AveragePrecision": 0.9876464664646313,
        "AverageRecall": 0.9445600253081214,
        "AverageAccuracy": 0.9997281665190434
      },
      "EvaluationManifestS3Prefix": "s3://DOC-EXAMPLE-BUCKET/example-flywheel/schemaVersion=1/20230619EXAMPLE/evaluation/20230619EXAMPLE/"
    },
    {
      "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/"
  }
]
}

```

Per ulteriori informazioni, consulta la [panoramica di Flywheel](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListFlywheelIterationHistory](#)Reference.

## list-flywheels

Il seguente esempio di codice mostra come utilizzare `list-flywheels`.

### AWS CLI

Per elencare tutti i volani

L'`list-flywheels`esempio seguente elenca tutti i volani creati.

```
aws comprehend list-flywheels
```

Output:

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta la [panoramica di Flywheel](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListFlywheels](#)Reference.

## list-key-phrases-detection-jobs

Il seguente esempio di codice mostra come utilizzare `list-key-phrases-detection-jobs`.

### AWS CLI

Per elencare tutte le frasi chiave, i lavori di rilevamento.

L'`list-key-phrases-detection-jobs`esempio seguente elenca tutti i processi di rilevamento delle frasi chiave asincroni in corso e completati.



**aws comprehend list-key-phrases-detection-jobs**

## Output:

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [ListKeyPhrasesDetectionJobs](#) Command Reference.AWS CLI

## list-pii-entities-detection-jobs

Il seguente esempio di codice mostra come utilizzare `list-pii-entities-detection-jobs`.

### AWS CLI

Per elencare tutti i processi di rilevamento delle entità pii

L'`list-pii-entities-detection-jobs`esempio seguente elenca tutti i processi di rilevamento pii asincroni in corso e completati.

**aws comprehend list-pii-entities-detection-jobs**

## Output:

```
{
  "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"
    }
]
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [ListPiiEntitiesDetectionJobs](#) Command Reference.AWS CLI

## list-sentiment-detection-jobs

Il seguente esempio di codice mostra come utilizzare `list-sentiment-detection-jobs`.

### AWS CLI

Per elencare tutti i lavori di rilevamento del sentiment

L'`list-sentiment-detection-jobs`esempio seguente elenca tutti i processi asincroni di rilevamento del sentiment in corso e completati.

```
aws comprehend list-sentiment-detection-jobs
```

Output:

```
{
  "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-123456abcdeb0e11022f22a1EXAMPLE/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  },
  {
    "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-
detection-job/123456abcdeb0e11022f22a1EXAMPLE2",
    "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-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
    },
    "LanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
  }
]
}

```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [ListSentimentDetectionJobs](#) Command Reference.AWS CLI

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag della risorsa

L'`list-tags-for-resource` seguente elenca i tag per una risorsa Amazon Comprehend.

```
aws comprehend list-tags-for-resource \
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
```

Output:

```
{
  "ResourceArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1",
  "Tags": [
    {
      "Key": "Department",
      "Value": "Finance"
    },
    {
      "Key": "location",
      "Value": "Seattle"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Tagging your resources](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListTagsForResource](#) Reference.

## **list-targeted-sentiment-detection-jobs**

Il seguente esempio di codice mostra come utilizzare `list-targeted-sentiment-detection-jobs`.

AWS CLI

Per elencare tutti i lavori mirati di rilevamento del sentiment

L'`list-targeted-sentiment-detection-jobs` seguente elenca tutti i lavori asincroni mirati di rilevamento del sentiment in corso e completati.

**aws comprehend list-targeted-sentiment-detection-jobs**

## Output:

```
{
  "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-123456abcdeb0e11022f22a11EXAMPLE2/output/output.tar.gz"
      },
      "LanguageCode": "en",
    }
  ]
}
```

```

        "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
    }
]
}

```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [ListTargetedSentimentDetectionJobs](#) Command Reference.AWS CLI

## list-topics-detection-jobs

Il seguente esempio di codice mostra come utilizzare `list-topics-detection-jobs`.

### AWS CLI

Per elencare tutti i lavori di rilevamento degli argomenti

L'`list-topics-detection-jobs` esempio seguente elenca tutti i processi di rilevamento di argomenti asincroni in corso e completati.

```
aws comprehend list-topics-detection-jobs
```

Output:

```

{
  "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-123456abcdeb0e11022f22a11EXAMPLE/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"
  }
}
```

```
]
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [ListTopicsDetectionJobs](#) Command Reference.AWS CLI

## put-resource-policy

Il seguente esempio di codice mostra come utilizzare `put-resource-policy`.

### AWS CLI

Per allegare una politica basata sulle risorse

L'`put-resource-policy`esempio seguente allega una politica basata sulle risorse a un modello in modo che possa essere importato da un altro account. AWS La policy è allegata al modello in account 111122223333 e consente all'account di importare il modello. 444455556666

```
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"]} ]}'
```

Uscita:

```
{
  "PolicyRevisionId": "aaa111d069d07afaa2aa3106aEXAMPLE"
}
```

Per ulteriori informazioni, [consulta Copiare modelli personalizzati tra AWS account](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [PutResourcePolicy](#)Reference.

## start-document-classification-job

Il seguente esempio di codice mostra come utilizzare `start-document-classification-job`.

## AWS CLI

Per iniziare un lavoro di classificazione dei documenti

L'`start-document-classification-job` seguente avvia un processo di classificazione dei documenti con un modello personalizzato su tutti i file all'indirizzo specificato dal `--input-data-config` tag. In questo esempio, il bucket S3 di input contiene `SampleSMStext1.txt`, `SampleSMStext2.txt`, e `SampleSMStext3.txt`. Il modello è stato precedentemente addestrato sulla classificazione dei documenti dei messaggi spam e non spam, o «ham». SMS. Quando il lavoro è completo, `output.tar.gz` viene inserito nella posizione specificata dal `--output-data-config` tag. `output.tar.gz` contiene `predictions.jsonl` che elenca la classificazione di ogni documento. L'output JSON viene stampato su una riga per file, ma è formattato qui per garantire la leggibilità.

```
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
```

Contenuto di `SampleSMStext1.txt`.

```
"CONGRATULATIONS! TXT 2155550100 to win $5000"
```

Contenuto di `SampleSMStext2.txt`.

```
"Hi, when do you want me to pick you up from practice?"
```

Contenuto di `SampleSMStext3.txt`.

```
"Plz send bank account # to 2155550100 to claim prize!!"
```

Output:

```
{  
  "JobId": "e758dd56b824aa717ceab551fEXAMPLE",
```

```

    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-classification-
    job/e758dd56b824aa717ceab551fEXAMPLE",
    "JobStatus": "SUBMITTED"
  }

```

Contenuto di `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}]}

```

Per ulteriori informazioni, consulta la sezione [Classificazione personalizzata](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [StartDocumentClassificationJob AWS CLI Command Reference](#).

## start-dominant-language-detection-job

Il seguente esempio di codice mostra come utilizzare `start-dominant-language-detection-job`.

### AWS CLI

Per avviare un processo asincrono di rilevamento della lingua

L'`start-dominant-language-detection-job` seguente avvia un processo asincrono di rilevamento della lingua per tutti i file che si trovano all'indirizzo specificato dal tag `--input-data-config`. Il bucket S3 in questo esempio contiene `Sampletext1.txt`. Quando il lavoro è completo, la cartella `output`, viene posizionata nella posizione specificata dal `--output-data-config` tag. La cartella `output.txt` contiene la lingua dominante di ogni file di testo e il punteggio di affidabilità del modello pre-addestrato per ogni previsione.

```

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
```

Contenuto di 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."
```

Output:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Contenuto di output.txt.

```
{"File": "Sampletext1.txt", "Languages": [{"LanguageCode": "en", "Score": 0.9913753867149353}], "Line": 0}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [StartDominantLanguageDetectionJob](#) Command Reference.AWS CLI

## start-entities-detection-job

Il seguente esempio di codice mostra come utilizzare `start-entities-detection-job`.

AWS CLI

Esempio 1: per avviare un processo standard di rilevamento delle entità utilizzando il modello pre-addestrato

L'`start-entities-detection-job` seguente avvia un processo di rilevamento asincrono delle entità per tutti i file che si trovano all'indirizzo specificato dal tag. `--input-data-config` Il bucket S3 in questo esempio contiene, e. `Sampletext1.txt` `Sampletext2.txt`

Sampletext3.txt Quando il lavoro è completo, la cartella, output, viene posizionata nella posizione specificata dal `--output-data-config` tag. La cartella `output.txt` contiene un elenco di tutte le entità denominate rilevate in ogni file di testo, nonché il punteggio di affidabilità del modello pre-addestrato per ogni previsione. L'output Json viene stampato su una riga per file di input, ma qui è formattato per motivi di leggibilità.

```
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
```

Contenuto di 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."
```

Contenuto di 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. "
```

Contenuto di 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."
```

Output:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-  
job/123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "SUBMITTED"  
}
```

## Contenuto di output .txt con rientri di riga per una maggiore leggibilità:

```
{
  "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
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

Esempio 2: per avviare un processo personalizzato di rilevamento delle entità

L'`start-entities-detection-job` seguente avvia un processo asincrono di rilevamento delle entità personalizzate per tutti i file che si trovano all'indirizzo specificato dal tag `--input-data-config`. In questo esempio, il bucket S3 di questo esempio contiene, e. `SampleFeedback1.txt` `SampleFeedback2.txt` `SampleFeedback3.txt`. Il modello Entity Recognizer è stato addestrato sulla base dei feedback dell'assistenza clienti per riconoscere i nomi dei dispositivi. Una volta completato il lavoro, la cartella `output`, viene inserita nella posizione specificata dal `--output-data-config` tag. La cartella `output.txt`, in cui sono elencate tutte le entità denominate rilevate in ogni file di testo, nonché il punteggio di affidabilità del modello pre-addestrato per ogni previsione. L'output Json viene stampato su una riga per file, ma qui è formattato per motivi di leggibilità.

```
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"
```

Contenuto di `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!"
```

Contenuto di `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!"
```

Contenuto di `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!"
```

Output:

```
{
  "JobId": "019ea9edac758806850fa8a79ff83021",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/019ea9edac758806850fa8a79ff83021",
  "JobStatus": "SUBMITTED"
}
```

Contenuto di output .txt con rientri di riga per una maggiore leggibilità:

```
{
  "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
}

```

Per ulteriori informazioni, consulta il [riconoscimento personalizzato delle entità](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [StartEntitiesDetectionJob AWS CLI Command Reference](#).

## start-events-detection-job

Il seguente esempio di codice mostra come utilizzare `start-events-detection-job`.

### AWS CLI

Per avviare un processo asincrono di rilevamento degli eventi

L'esempio seguente avvia un processo asincrono di rilevamento degli eventi per tutti i file che si trovano all'indirizzo specificato dal tag. `--input-data-config` I possibili tipi di eventi di destinazione includono `BANKRUPTCYEMPLOYMENT`, `CORPORATE_ACQUISITION`, `INVESTMENT_GENERAL`, `CORPORATE_MERGER`, `SECONDARY_OFFERINGS`, `SHELF_OFFERING`, `TENDER_OFFERING` e `STOCK_SPLIT`. Il bucket S3 in questo esempio contiene `SampleText1.txt`, `SampleText2.txt`, e `SampleText3.txt`. Quando il lavoro è completo, la cartella, `output`, viene posizionata nella posizione specificata dal `--output-data-config` tag. La cartella contiene `SampleText1.txt.out`, `SampleText2.txt.out`, e `SampleText3.txt.out`. L'JSON output viene stampato su una riga per file, ma qui è formattato per motivi di leggibilità.

```

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

```

Contenuto di `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."
```

Contenuto di SampleText2.txt.

```
"In 2021, AnyCompany officially purchased AnyBusiness for 100 billion dollars, surprising and exciting the shareholders."
```

Contenuto di SampleText3.txt.

```
"In 2022, AnyCompany stock crashed 50. Eventually later that year they filed for bankruptcy."
```

Output:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:events-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Contenuto di SampleText1.txt.out con rientri di riga per una maggiore leggibilità:

```
{
  "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
            }
        ],
        "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
}
```

### Contenuto di 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
}

```

Contenuto di 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
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [StartEventsDetectionJob](#) Command Reference.AWS CLI

## start-flywheel-iteration

Il seguente esempio di codice mostra come utilizzare `start-flywheel-iteration`.

### AWS CLI

Per avviare un'iterazione del volano

L'`start-flywheel-iteration`esempio seguente avvia un'iterazione del volano. Questa operazione utilizza qualsiasi nuovo set di dati nel volano per addestrare una nuova versione del modello.

```
aws comprehend start-flywheel-iteration \
  --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel
```

Output:

```
{
  "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel",
  "FlywheelIterationId": "12345123TEXAMPLE"
}
```

Per ulteriori informazioni, consulta la [panoramica di Flywheel](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [StartFlywheelIteration](#) Reference.

## start-key-phrases-detection-job

Il seguente esempio di codice mostra come utilizzare `start-key-phrases-detection-job`.

### AWS CLI

Per avviare un processo di rilevamento di frasi chiave

L'esempio seguente avvia un processo di rilevamento asincrono delle frasi chiave per tutti i file che si trovano all'indirizzo specificato dal tag `--input-data-config`. Il bucket S3 in questo esempio contiene, e. `Sampletext1.txt`, `Sampletext2.txt`, `Sampletext3.txt`. Una volta completato il lavoro, la cartella `output`, viene posizionata nella posizione specificata dal `--output-data-config` tag. La cartella contiene il file `output.txt` che contiene tutte le frasi chiave rilevate in ogni file di testo e il punteggio di affidabilità del modello pre-addestrato per ogni previsione. L'output JSON viene stampato su una riga per file, ma è formattato qui per garantire la leggibilità.

```
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
```

Contenuto di `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."
```

Contenuto di `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. "
```

Contenuto di `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."
```

### Output:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

### Contenuto di output.txt con rientri di riga per la leggibilità:

```
{
  "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  
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [StartKeyPhrasesDetectionJob](#) Command Reference.AWS CLI

## **start-pii-entities-detection-job**

Il seguente esempio di codice mostra come utilizzare `start-pii-entities-detection-job`.

## AWS CLI

Per avviare un processo di rilevamento asincrono PII

L'`start-pii-entities-detection-job` seguente avvia un processo asincrono di rilevamento delle entità di informazioni di identificazione personale (PII) per tutti i file che si trovano all'indirizzo specificato dal tag `--input-data-config` Il bucket S3 in questo esempio contiene, e. `Sampletext1.txt` `Sampletext2.txt` `Sampletext3.txt` Quando il lavoro è completo, la cartella, `output`, viene posizionata nella posizione specificata dal `--output-data-config` tag. La cartella contiene `SampleText1.txt.out` e elenca `SampleText3.txt.out` le entità denominate all'interno di ogni file di testo. `SampleText2.txt.out` L'output Json viene stampato su una riga per file, ma qui è formattato per motivi di leggibilità.

```
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
```

Contenuto di `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."
```

Contenuto di `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. "
```

Contenuto di `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."
```

Output:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pII-entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "SUBMITTED"
}
```

Contenuto di `SampleText1.txt.out` con rientri di riga per una maggiore leggibilità:

```
{
  "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
}
```

Contenuto della confezione `SampleText2.txt.out` con rientri di riga per una maggiore leggibilità:

```
{
  "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
}
```

Contenuto della confezione SampleText3.txt.out con rientri di riga per una maggiore leggibilità:

```
{
  "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
}

```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [StartPiiEntitiesDetectionJob](#) Command Reference.AWS CLI

## start-sentiment-detection-job

Il seguente esempio di codice mostra come utilizzare `start-sentiment-detection-job`.

### AWS CLI

Per avviare un lavoro asincrono di analisi del sentiment

L'esempio seguente avvia un processo asincrono di rilevamento dell'analisi del sentiment per tutti i file che si trovano all'indirizzo specificato dal tag. `--input-data-config` La cartella bucket S3 in questo esempio contiene, e. `SampleMovieReview1.txt SampleMovieReview2.txt SampleMovieReview3.txt` Quando il lavoro è completo, la cartella, `output`, viene posizionata nella posizione specificata dal `--output-data-config` tag. La cartella contiene il file `output.txt`, che contiene i sentimenti prevalenti per ogni file di testo e il punteggio di affidabilità del modello pre-addestrato per ogni

previsione. L'output Json viene stampato su una riga per file, ma qui è formattato per garantire la leggibilità.

```
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
```

Contenuto di SampleMovieReview1.txt.

```
"The film, AnyMovie2, is fairly predictable and just okay."
```

Contenuto di SampleMovieReview2.txt.

```
"AnyMovie2 is the essential sci-fi film that I grew up watching when I was a kid. I  
highly recommend this movie."
```

Contenuto di SampleMovieReview3.txt.

```
"Don't get fooled by the 'awards' for AnyMovie2. All parts of the film were poorly  
stolen from other modern directors."
```

Output:

```
{  
  "JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-detection-  
job/0b5001e25f62ebb40631a9a1a7fde7b3",  
  "JobStatus": "SUBMITTED"  
}
```

Contenuto di output.txt con riga di rientri per garantire la leggibilità:

```
{  
  "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
  }
}
}

```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [StartSentimentDetectionJob](#) Command Reference.AWS CLI

## start-targeted-sentiment-detection-job

Il seguente esempio di codice mostra come utilizzare `start-targeted-sentiment-detection-job`.

### AWS CLI

Per avviare un lavoro asincrono di analisi mirata del sentiment

L'`start-targeted-sentiment-detection-job` seguente avvia un processo asincrono di rilevamento mirato del sentiment per tutti i file che si trovano all'indirizzo specificato dal tag. `--input-data-config` La cartella bucket S3 in questo esempio contiene, e. `SampleMovieReview1.txt` `SampleMovieReview2.txt` `SampleMovieReview3.txt` Quando il lavoro è completo, `output.tar.gz` viene posizionato nella posizione specificata dal `--output-data-config` tag. `output.tar.gz` contiene i file e `SampleMovieReview1.txt.out` `SampleMovieReview2.txt.out` `SampleMovieReview3.txt.out`, ciascuno dei quali contiene tutte le entità denominate e i sentimenti associati per un singolo file di testo di input.

```
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
```

Contenuto di `SampleMovieReview1.txt`.

```
"The film, AnyMovie, is fairly predictable and just okay."
```

Contenuto di `SampleMovieReview2.txt`.

```
"AnyMovie is the essential sci-fi film that I grew up watching when I was a kid. I  
highly recommend this movie."
```

Contenuto di `SampleMovieReview3.txt`.

```
"Don't get fooled by the 'awards' for AnyMovie. All parts of the film were poorly  
stolen from other modern directors."
```

Output:

```
{  
  "JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",  
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-  
detection-job/0b5001e25f62ebb40631a9a1a7fde7b3",  
  "JobStatus": "SUBMITTED"
```



```
}
```

Contenuto di SampleMovieReview1.txt.out con rientri a riga per una maggiore leggibilità:

```
{
  "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
}

```

Contenuto dei rientri di SampleMovieReview2.txt.out riga per motivi di leggibilità:

```

{
  "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
}

```

Contenuto della confezione SampleMovieReview3.txt.out con rientri di riga per garantire la leggibilità:

```

{
  "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
  }

```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [StartTargetedSentimentDetectionJob](#) Command Reference.AWS CLI

## start-topics-detection-job

Il seguente esempio di codice mostra come utilizzare `start-topics-detection-job`.

### AWS CLI

Per avviare un lavoro di analisi del rilevamento di argomenti

L'`start-topics-detection-job`esempio seguente avvia un processo asincrono di rilevamento degli argomenti per tutti i file che si trovano all'indirizzo specificato dal tag. `--input-data-config` Quando il processo è completo, la cartella, `output`, viene posizionata nella posizione specificata dal tag. `--ouput-data-config` `output`contiene `topic-terms.csv` e `doc-topics.csv`. Il primo file di output, `topic-terms.csv`, è un elenco di argomenti della raccolta. Per ogni argomento, l'elenco include, per impostazione predefinita, i termini principali per argomento in base al loro peso. Il secondo file elenca i documenti associati a un argomento e la proporzione del documento che riguarda l'argomento. `doc-topics.csv`

```

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

```

Output:

```

{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-job/123456abcdeb0e11022f22a11EXAMPLE",

```



```
"JobStatus": "SUBMITTED"
}
```

Per ulteriori informazioni, consulta [Topic Modeling](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [StartTopicsDetectionJob](#)Reference.

## stop-dominant-language-detection-job

Il seguente esempio di codice mostra come utilizzare `stop-dominant-language-detection-job`.

### AWS CLI

Per interrompere un processo asincrono di rilevamento della lingua dominante

L'`stop-dominant-language-detection-job`esempio seguente interrompe un processo di rilevamento asincrono della lingua dominante in corso. Se lo stato corrente del lavoro è, `IN_PROGRESS` il lavoro viene contrassegnato come terminato e inserito nello stato. `STOP_REQUESTED` Se il processo viene completato prima di poter essere interrotto, viene messo nello `COMPLETED` stato.

```
aws comprehend stop-dominant-language-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Output:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend](#) Developer Guide.

- Per i API dettagli, consulta [StopDominantLanguageDetectionJob](#)Command Reference.AWS CLI

## stop-entities-detection-job

Il seguente esempio di codice mostra come utilizzare `stop-entities-detection-job`.

## AWS CLI

Per interrompere un processo di rilevamento di entità asincrone

L'`stop-entities-detection-job` seguente interrompe un processo di rilevamento di entità asincrone in corso. Se lo stato corrente del lavoro è, `IN_PROGRESS` il lavoro viene contrassegnato come terminato e inserito nello stato. `STOP_REQUESTED` Se il processo viene completato prima di poter essere interrotto, viene messo nello `COMPLETED` stato.

```
aws comprehend stop-entities-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Output:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "STOP_REQUESTED"  
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [StopEntitiesDetectionJob](#) Command Reference.AWS CLI

## `stop-events-detection-job`

Il seguente esempio di codice mostra come utilizzare `stop-events-detection-job`.

### AWS CLI

Per interrompere un processo asincrono di rilevamento degli eventi

L'`stop-events-detection-job` seguente interrompe un processo di rilevamento di eventi asincrono in corso. Se lo stato corrente del lavoro è, `IN_PROGRESS` il lavoro viene contrassegnato per l'interruzione e inserito nello stato. `STOP_REQUESTED` Se il processo viene completato prima di poter essere interrotto, viene messo nello `COMPLETED` stato.

```
aws comprehend stop-events-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Output:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [StopEventsDetectionJob](#) Command Reference.AWS CLI

## stop-key-phrases-detection-job

Il seguente esempio di codice mostra come utilizzare `stop-key-phrases-detection-job`.

AWS CLI

Per interrompere un processo di rilevamento asincrono di frasi chiave

L'`stop-key-phrases-detection-job` esempio seguente interrompe un processo di rilevamento di frasi chiave asincrone in corso. Se lo stato corrente del lavoro è, `IN_PROGRESS` il lavoro viene contrassegnato come terminato e inserito nello stato. `STOP_REQUESTED` Se il processo viene completato prima di poter essere interrotto, viene messo nello `COMPLETED` stato.

```
aws comprehend stop-key-phrases-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Output:

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [StopKeyPhrasesDetectionJob](#) Command Reference.AWS CLI

## stop-pii-entities-detection-job

Il seguente esempio di codice mostra come utilizzare `stop-pii-entities-detection-job`.

## AWS CLI

Per interrompere un processo asincrono di rilevamento di entità pii

L'`stop-pii-entities-detection-job` seguente interrompe un processo di rilevamento di entità pii asincrone in corso. Se lo stato corrente del lavoro è, `IN_PROGRESS` il lavoro viene contrassegnato come terminato e inserito nello stato. `STOP_REQUESTED` Se il processo viene completato prima di poter essere interrotto, viene messo nello `COMPLETED` stato.

```
aws comprehend stop-pii-entities-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

Output:

```
{  
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",  
  "JobStatus": "STOP_REQUESTED"  
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [StopPiiEntitiesDetectionJob](#) Command Reference.AWS CLI

## `stop-sentiment-detection-job`

Il seguente esempio di codice mostra come utilizzare `stop-sentiment-detection-job`.

## AWS CLI

Per interrompere un processo asincrono di rilevamento del sentiment

L'`stop-sentiment-detection-job` seguente interrompe un processo asincrono di rilevamento del sentiment in corso. Se lo stato corrente del lavoro è, `IN_PROGRESS` il lavoro viene contrassegnato come terminato e inserito nello stato. `STOP_REQUESTED` Se il processo viene completato prima di poter essere interrotto, viene messo nello `COMPLETED` stato.

```
aws comprehend stop-sentiment-detection-job \  
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

**Output:**

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [StopSentimentDetectionJob](#) Command Reference.AWS CLI

**stop-targeted-sentiment-detection-job**

Il seguente esempio di codice mostra come utilizzare `stop-targeted-sentiment-detection-job`.

**AWS CLI**

Per interrompere un processo asincrono di rilevamento mirato del sentiment

L'`stop-targeted-sentiment-detection-job`esempio seguente interrompe un processo asincrono di rilevamento mirato del sentiment in corso. Se lo stato corrente del lavoro è, `IN_PROGRESS` il lavoro viene contrassegnato come terminato e inserito nello stato. `STOP_REQUESTED` Se il processo viene completato prima di poter essere interrotto, viene messo nello `COMPLETED` stato.

```
aws comprehend stop-targeted-sentiment-detection-job \
  --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

**Output:**

```
{
  "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
  "JobStatus": "STOP_REQUESTED"
}
```

Per ulteriori informazioni, consulta l'[analisi asincrona per Amazon Comprehend Insights nella Amazon Comprehend Developer Guide](#).

- Per i API dettagli, consulta [StopTargetedSentimentDetectionJob](#) Command Reference.AWS CLI

## stop-training-document-classifier

Il seguente esempio di codice mostra come utilizzare `stop-training-document-classifier`.

### AWS CLI

Per interrompere l'addestramento di un modello di classificazione dei documenti

L'`stop-training-document-classifier` seguente interrompe l'addestramento di un modello di classificatore di documenti in corso.

```
aws comprehend stop-training-document-classifier
  --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-
  classifier/example-classifier
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creazione e gestione di modelli personalizzati](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [StopTrainingDocumentClassifier AWS CLI Command Reference](#).

## stop-training-entity-recognizer

Il seguente esempio di codice mostra come utilizzare `stop-training-entity-recognizer`.

### AWS CLI

Per interrompere l'addestramento di un modello di riconoscimento delle entità

L'`stop-training-entity-recognizer` seguente interrompe l'addestramento di un modello di riconoscimento delle entità mentre è in corso.

```
aws comprehend stop-training-entity-recognizer
  --entity-recognizer-arn "arn:aws:comprehend:us-west-2:111122223333:entity-
  recognizer/examplerecognizer1"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creazione e gestione di modelli personalizzati](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta [StopTrainingEntityRecognizer AWS CLI Command Reference](#).

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Esempio 1: etichettare una risorsa

L'`tag-resource`esempio seguente aggiunge un singolo tag a una risorsa Amazon Comprehend.

```
aws comprehend tag-resource \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier/version/1 \  
  --tags Key=Location,Value=Seattle
```

Questo comando non ha alcun output.

Per ulteriori informazioni, consulta [Tagging your resources](#) nella Amazon Comprehend Developer Guide.

Esempio 2: per aggiungere più tag a una risorsa

L'`tag-resource`esempio seguente aggiunge più tag a una risorsa Amazon Comprehend.

```
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
```

Questo comando non ha alcun output.

Per ulteriori informazioni, consulta [Tagging your resources](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [TagResource](#)Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

## AWS CLI

Esempio 1: rimuovere un singolo tag da una risorsa

L'untag-resourceesempio seguente rimuove un singolo tag da una risorsa Amazon Comprehend.

```
aws comprehend untag-resource \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier/version/1 \  
  --tag-keys Location
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging your resources](#) nella Amazon Comprehend Developer Guide.

Esempio 2: rimuovere più tag da una risorsa

L'untag-resourceesempio seguente rimuove più tag da una risorsa Amazon Comprehend.

```
aws comprehend untag-resource \  
  --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/  
example-classifier/version/1 \  
  --tag-keys Location Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging your resources](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UntagResource](#)Reference.

## update-endpoint

Il seguente esempio di codice mostra come utilizzareupdate-endpoint.

## AWS CLI

Esempio 1: aggiornare le unità di inferenza di un endpoint



L'update-endpointesempio seguente aggiorna le informazioni su un endpoint. In questo esempio, il numero di unità di inferenza viene aumentato.

```
aws comprehend update-endpoint \  
  --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-  
endpoint/example-classifier-endpoint \  
  --desired-inference-units 2
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing Amazon Comprehend endpoint nella Amazon Comprehend Developer Guide](#).

Esempio 2: aggiornare il modello attivo di un endpoint

L'update-endpointesempio seguente aggiorna le informazioni su un endpoint. In questo esempio, il modello attivo viene modificato.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing Amazon Comprehend endpoint nella Amazon Comprehend Developer Guide](#).

- Per API i dettagli, consulta Command [UpdateEndpointReference AWS CLI](#) .

## update-flywheel

Il seguente esempio di codice mostra come utilizzareupdate-flywheel.

### AWS CLI

Per aggiornare una configurazione del volano

L'update-flywheel esempio seguente aggiorna una configurazione del volano. In questo esempio, il modello attivo per il volano viene aggiornato.

```
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
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta la [panoramica di Flywheel](#) nella Amazon Comprehend Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateFlywheel](#) Reference.

## Esempi di utilizzo di Amazon Comprehend Medical AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Comprehend Medical. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **describe-entities-detection-v2-job**

Il seguente esempio di codice mostra come utilizzare `describe-entities-detection-v2-job`.

#### AWS CLI

Per descrivere un processo di rilevamento delle entità

L'`describe-entities-detection-v2-job`esempio seguente visualizza le proprietà associate a un processo asincrono di rilevamento delle entità.

```
aws comprehendmedical describe-entities-detection-v2-job \  
--job-id "ab9887877365fe70299089371c043b96"
```

#### Output:

```
{  
  "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"
}
}

```

Per ulteriori informazioni, consulta [Batch APIs](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta [DescribeEntitiesDetectionV2Job](#) in AWS CLI Command Reference.

## describe-icd10-cm-inference-job

Il seguente esempio di codice mostra come utilizzare `describe-icd10-cm-inference-job`

### AWS CLI

Per descrivere un lavoro di ICD inferenza di -10 CM

L'`describe-icd10-cm-inference-job` seguente descrive le proprietà del lavoro di inferenza richiesto con il job-id specificato.

```

aws comprehendmedical describe-icd10-cm-inference-job \
  --job-id "5780034166536cdb52ffa3295a1b00a7"

```

Output:

```

{
  "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"
}
}

```

Per ulteriori informazioni, consulta [Ontology linking batch analysis](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta la sezione [DescribeCommandInferenceJob](#) in AWS CLI Command Reference.

## describe-phi-detection-job

Il seguente esempio di codice mostra come utilizzare `describe-phi-detection-job`.

### AWS CLI

Per descrivere un processo PHI di rilevamento

L'esempio seguente visualizza le proprietà associate a un processo asincrono di rilevamento delle informazioni sanitarie protette (PHI).

```

aws comprehendmedical describe-phi-detection-job \
  --job-id "4750034166536cdb52ffa3295a1b00a3"

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Batch APIs](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta [DescribePhiDetectionJob AWS CLI Command Reference](#).

## describe-rx-norm-inference-job

Il seguente esempio di codice mostra come utilizzare `describe-rx-norm-inference-job`.

### AWS CLI

Per descrivere un lavoro di RxNorm inferenza

L'`describe-rx-norm-inference-job` esempio seguente descrive le proprietà del lavoro di inferenza richiesto con il job-id specificato.

```

aws comprehendmedical describe-rx-norm-inference-job \
  --job-id "eg8199877365fc70299089371c043b96"

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Ontology linking batch analysis](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta Command [DescribeRxNormInferenceJob](#)Reference AWS CLI .

## describe-snomedct-inference-job

Il seguente esempio di codice mostra come utilizzare `describe-snomedct-inference-job`.

### AWS CLI

Per descrivere un lavoro di inferenza SNOMED CT

L'`describe-snomedct-inference-job`esempio seguente descrive le proprietà del lavoro di inferenza richiesto con il job-id specificato.

```

aws comprehendmedical describe-snomedct-inference-job \
  --job-id "2630034166536cdb52ffa3295a1b00a7"

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Ontology linking batch analysis](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta Command [DescribeSnomedctInferenceJob](#) Reference AWS CLI .

## detect-entities-v2

Il seguente esempio di codice mostra come utilizzare `detect-entities-v2`.

### AWS CLI

Esempio 1: per rilevare le entità direttamente dal testo

L'esempio seguente mostra le entità rilevate e le etichetta in base al tipo, direttamente dal testo di input.

```

aws comprehendmedical detect-entities-v2 \
  --text "Sleeping trouble on present dosage of Clonidine. Severe rash on face and leg, slightly itchy."

```

Output:

```

{
  "Id": 0,
  "BeginOffset": 38,
  "EndOffset": 47,
  "Score": 0.9942955374717712,
  "Text": "Clonidine",
  "Category": "MEDICATION",
  "Type": "GENERIC_NAME",
  "Traits": []
}

```



```
}
```

Per ulteriori informazioni, consulta [Detect Entities versione 2](#) nella Amazon Comprehend Medical Developer Guide.

Esempio 2: per rilevare entità da un percorso di file

L'`detect-entities-v2`esempio seguente mostra le entità rilevate e le etichetta in base al tipo di un percorso di file.

```
aws comprehendmedical detect-entities-v2 \  
  --text file://medical_entities.txt
```

Contenuto di `medical_entities.txt`.

```
{  
  "Sleeping trouble on present dosage of Clonidine. Severe rash on face and leg,  
  slightly itchy."  
}
```

Output:

```
{  
  "Id": 0,  
  "BeginOffset": 38,  
  "EndOffset": 47,  
  "Score": 0.9942955374717712,  
  "Text": "Clonidine",  
  "Category": "MEDICATION",  
  "Type": "GENERIC_NAME",  
  "Traits": []  
}
```

Per ulteriori informazioni, consulta [Detect Entities versione 2](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta [DetectEntitiesV2](#) in AWS CLI Command Reference.

## **detect-phi**

Il seguente esempio di codice mostra come utilizzare `detect-phi`.

## AWS CLI

Esempio 1: rilevare informazioni sanitarie protette (PHI) direttamente dal testo

L'output seguente visualizza le entità di informazioni sanitarie protette (PHI) rilevate direttamente dal testo di input.

```
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."
```

Output:

```
{  
  "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"
  }

```

Per ulteriori informazioni, consulta [Detect PHI](#) nella Amazon Comprehend Medical Developer Guide.

Esempio 2: rilevare le informazioni sanitarie protette (PHI) direttamente dal percorso di un file

L'output seguente mostra le entità protected health information (PHI) rilevate da un percorso di file.

```

aws comprehendmedical detect-phi \
  --text file://phi.txt

```

Contenuto di `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."

```

Output:

```

{
  "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"
}

```

Per ulteriori informazioni, consulta [Detect PHI](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta [DetectPhi AWS CLI Command Reference](#).

## **infer-icd10-cm**

Il seguente esempio di codice mostra come utilizzare `infer-icd10-cm`.

### AWS CLI

Esempio 1: per rilevare entità relative a condizioni mediche e collegarle all'ontologia ICD -10-CM direttamente dal testo

L'`infer-icd10-cm`esempio seguente etichetta le entità relative alle condizioni mediche rilevate e le collega ai codici dell'edizione 2019 della Classificazione internazionale delle malattie modificate cliniche (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."

```

## Output:

```
{
  "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"
}

```

Per ulteriori informazioni, consulta [Infer ICD1 0-CM](#) nella Amazon Comprehend Medical Developer Guide.

Esempio 2: rilevare entità affette da condizioni mediche e collegarsi all'ontologia ICD -10-CM da un percorso di file

L'`infer-icd-10-cmesempio` seguente etichetta le entità relative alle condizioni mediche rilevate e le collega ai codici dell'edizione 2019 della Classificazione internazionale delle malattie modificate cliniche (-10-CM). ICD

```

aws comprehendmedical infer-icd10-cm \
  --text file://icd10cm.txt

```

Contenuto di `icd10cm.txt`.

```

{
  "The patient complains of abdominal pain, has a long-standing history of
  diabetes treated with Micronase daily."
}

```

Output:

```

{
  "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"
}

```

Per ulteriori informazioni, consulta [ICD1Infer-0-CM](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta [Inferlcd10Cm](#) in Command Reference.AWS CLI

## infer-rx-norm

Il seguente esempio di codice mostra come utilizzare `infer-rx-norm`.

### AWS CLI

Esempio 1: per rilevare le entità del farmaco e collegarle RxNorm direttamente dal testo

L'`infer-rx-norm` esempio seguente mostra ed etichetta le entità terapeutiche rilevate e le collega agli identificatori concettuali (RxCUI) del database della 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."
```

Output:

```
{  
  "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"
}

```

Per ulteriori informazioni, consulta [Infer RxNorm](#) nella Amazon Comprehend Medical Developer Guide.

Esempio 2: per rilevare le entità del farmaco e collegarle RxNorm da un percorso di file.

L'infer-rx-normesempio seguente mostra ed etichetta le entità terapeutiche rilevate e le collega agli identificatori concettuali (RxCUI) del database della National Library of Medicine RxNorm .

```

aws comprehendmedical infer-rx-norm \
  --text file://rxnorm.txt

```

Contenuto di rxnorm.txt.

```

{
  "Patient reports taking Levothyroxine 125 micrograms p.o. once daily, but denies
  taking Synthroid."
}

```

Output:

```

{
  "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
[Levoxyl]",
          "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"
}

```

Per ulteriori informazioni, consulta [Infer RxNorm](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [InferRxNorm](#)Reference.

## **infer-snomedct**

Il seguente esempio di codice mostra come utilizzare `infer-snomedct`.

### AWS CLI

Esempio: per rilevare entità e collegarsi a SNOMED CT Ontology direttamente dal testo

L'`infer-snomedct` esempio seguente mostra come individuare le entità mediche e collegarle ai concetti della versione 2021-03 della nomenclatura sistematica della medicina, Clinical Terms (CT). SNOMED



```
aws comprehendmedical infer-snomedct \  
  --text "The patient complains of abdominal pain, has a long-standing history of  
diabetes treated with Micronase daily."
```

Output:

```
{  
  "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"
}

```

Per ulteriori informazioni, consulta [Infer SNOMEDCT](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [InferSnomedct](#)Reference.

## list-entities-detection-v2-jobs

Il seguente esempio di codice mostra come utilizzare `list-entities-detection-v2-jobs`.

### AWS CLI

Per elencare i lavori di rilevamento delle entità

L'`list-entities-detection-v2-jobs`esempio seguente elenca gli attuali processi di rilevamento asincrono.

```
aws comprehendmedical list-entities-detection-v2-jobs
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Batch APIs](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta [ListEntitiesDetectionV2Jobs](#) in AWS CLI Command Reference.

## list-icd10-cm-inference-jobs

Il seguente esempio di codice mostra come utilizzare `list-icd10-cm-inference-jobs`

### AWS CLI

Per elencare tutti gli attuali lavori di ICD inferenza di -10-CM

L'esempio seguente mostra come l'`list-icd10-cm-inference-jobs` operazione restituisce un elenco degli attuali processi di inferenza batch asincroni ICD -10-CM.

```
aws comprehendmedical list-icd10-cm-inference-jobs
```

**Output:**

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Ontology linking batch analysis](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta la sezione [ListId10 CmlInferenceJobs](#) in AWS CLI Command Reference.

**list-phi-detection-jobs**

Il seguente esempio di codice mostra come utilizzare `list-phi-detection-jobs`.

**AWS CLI**

Per elencare i lavori di rilevamento di informazioni sanitarie protette (PHI)

L'`list-phi-detection-jobs` seguente elenca gli attuali processi di rilevamento di informazioni sanitarie protette (PHI)

**aws comprehendmedical list-phi-detection-jobs**

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Batch APIs](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta [ListPhiDetectionJobs AWS CLI](#) Command Reference.

**list-rx-norm-inference-jobs**

Il seguente esempio di codice mostra come utilizzare `list-rx-norm-inference-jobs`.

**AWS CLI**

Per elencare tutti i lavori di inferenza Rx-Norm correnti

L'esempio seguente mostra come `list-rx-norm-inference-jobs` restituisce un elenco di lavori di inferenza batch Rx-Norm asincroni correnti.

```
aws comprehendmedical list-rx-norm-inference-jobs
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Ontology linking batch analysis](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta Command [ListRxNormInferenceJobs](#) Reference AWS CLI .

## **list-snomedct-inference-jobs**

Il seguente esempio di codice mostra come utilizzare `list-snomedct-inference-jobs`.

## AWS CLI

Per elencare tutti i SNOMED lavori di inferenza CT

L'esempio seguente mostra come l'`list-snomedct-inference-jobs` operazione restituisce un elenco degli attuali lavori di inferenza batch SNOMED CT asincroni.

```
aws comprehendmedical list-snomedct-inference-jobs
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Ontology linking batch analysis](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta Command [ListSnomedctInferenceJobs](#) Reference AWS CLI .

## **start-entities-detection-v2-job**

Il seguente esempio di codice mostra come utilizzare `start-entities-detection-v2-job`.

## AWS CLI

Per avviare un processo di rilevamento delle entità

L'`start-entities-detection-v2-job` seguente avvia un processo asincrono di rilevamento delle entità.

```
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
```

Output:

```
{  
  "JobId": "ab9887877365fe70299089371c043b96"  
}
```

Per ulteriori informazioni, consulta [Batch APIs](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta [StartEntitiesDetectionV2Job](#) in AWS CLI Command Reference.

## `start-icd10-cm-inference-job`

Il seguente esempio di codice mostra come utilizzare `start-icd10-cm-inference-job`

### AWS CLI

Per avviare un processo di ICD inferenza di -10 CM

L'`start-icd10-cm-inference-job` seguente avvia un processo di analisi in batch di ICD inferenza da 10 cm.

```
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
```



```
--language-code en
```

Output:

```
{  
  "JobId": "ef7289877365fc70299089371c043b96"  
}
```

Per ulteriori informazioni, consulta [Ontology linking batch analysis](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta la sezione [StartIcd10 CmInferenceJob](#) in AWS CLI Command Reference.

## start-phi-detection-job

Il seguente esempio di codice mostra come utilizzare `start-phi-detection-job`.

### AWS CLI

Per avviare un processo PHI di rilevamento

L'`start-phi-detection-job`esempio seguente avvia un processo asincrono di rilevamento delle PHI entità.

```
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
```

Output:

```
{  
  "JobId": "ab9887877365fe70299089371c043b96"  
}
```

Per ulteriori informazioni, consulta [Batch APIs](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta [StartPhiDetectionJob AWS CLI Command Reference](#).

## start-rx-norm-inference-job

Il seguente esempio di codice mostra come utilizzare `start-rx-norm-inference-job`.

### AWS CLI

Per avviare un processo di RxNorm inferenza

L'`start-rx-norm-inference-job`esempio seguente avvia un processo di analisi in batch di RxNorm inferenza.

```
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
```

Output:

```
{  
  "JobId": "eg8199877365fc70299089371c043b96"  
}
```

Per ulteriori informazioni, consulta [Ontology linking batch analysis](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta Command [StartRxNormInferenceJob](#)Reference AWS CLI .

## start-snomedct-inference-job

Il seguente esempio di codice mostra come utilizzare `start-snomedct-inference-job`.

### AWS CLI

Per avviare un processo di inferenza SNOMED CT

L'`start-snomedct-inference-job`esempio seguente avvia un processo di analisi in batch di inferenza SNOMED CT.

```
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
```

Output:

```
{  
  "JobId": "dg7289877365fc70299089371c043b96"  
}
```

Per ulteriori informazioni, consulta [Ontology linking batch analysis](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta Command [StartSnomedctInferenceJob](#) Reference AWS CLI .

## stop-entities-detection-v2-job

Il seguente esempio di codice mostra come utilizzare `stop-entities-detection-v2-job`.

AWS CLI

Per interrompere un processo di rilevamento delle entità

L'`stop-entities-detection-v2-job` esempio seguente interrompe un processo asincrono di rilevamento delle entità.

```
aws comprehendmedical stop-entities-detection-v2-job \  
  --job-id "ab9887877365fe70299089371c043b96"
```

Output:

```
{  
  "JobId": "ab9887877365fe70299089371c043b96"  
}
```

Per ulteriori informazioni, consulta [Batch APIs](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta [StopEntitiesDetectionV2Job](#) in AWS CLI Command Reference.

## stop-icd10-cm-inference-job

Il seguente esempio di codice mostra come utilizzare `stop-icd10-cm-inference-job`

### AWS CLI

Per interrompere un processo di ICD inferenza di -10 CM

L'`stop-icd10-cm-inference-job` seguente interrompe un processo di analisi in batch di ICD inferenza da 10 cm.

```
aws comprehendmedical stop-icd10-cm-inference-job \  
  --job-id "4750034166536cdb52ffa3295a1b00a3"
```

Output:

```
{  
  "JobId": "ef7289877365fc70299089371c043b96",  
}
```

Per ulteriori informazioni, consulta [Ontology linking batch analysis](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta la sezione [StopIcd10 CmInferenceJob](#) in AWS CLI Command Reference.

## stop-phi-detection-job

Il seguente esempio di codice mostra come utilizzare `stop-phi-detection-job`.

### AWS CLI

Per interrompere un processo di rilevamento di informazioni sanitarie protette (PHI)

L'`stop-phi-detection-job` seguente interrompe un processo asincrono di rilevamento di informazioni sanitarie protette (PHI).

```
aws comprehendmedical stop-phi-detection-job \  
  --job-id "4750034166536cdb52ffa3295a1b00a3"
```

```
--job-id "4750034166536cdb52ffa3295a1b00a3"
```

Output:

```
{
  "JobId": "ab9887877365fe70299089371c043b96"
}
```

Per ulteriori informazioni, consulta [Batch APIs](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta [StopPhiDetectionJob AWS CLI Command Reference](#).

## stop-rx-norm-inference-job

Il seguente esempio di codice mostra come utilizzare `stop-rx-norm-inference-job`.

AWS CLI

Per interrompere un processo di RxNorm inferenza

L'`stop-rx-norm-inference-job` esempio seguente interrompe un processo di analisi ICD in batch di inferenza di 10 cm.

```
aws comprehendmedical stop-rx-norm-inference-job \
  --job-id "eg8199877365fc70299089371c043b96"
```

Output:

```
{
  "JobId": "eg8199877365fc70299089371c043b96",
}
```

Per ulteriori informazioni, consulta [Ontology linking batch analysis](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta Command [StopRxNormInferenceJob](#) Reference AWS CLI .

## stop-snomedct-inference-job

Il seguente esempio di codice mostra come utilizzare `stop-snomedct-inference-job`.

## AWS CLI

Per interrompere un processo di inferenza SNOMED CT

L'`stop-snomedct-inference-job` seguente interrompe un processo di analisi in batch di inferenza SNOMED CT.

```
aws comprehendmedical stop-snomedct-inference-job \  
  --job-id "8750034166436cdb52ffa3295a1b00a1"
```

Output:

```
{  
  "JobId": "8750034166436cdb52ffa3295a1b00a1",  
}
```

Per ulteriori informazioni, consulta [Ontology linking batch analysis](#) nella Amazon Comprehend Medical Developer Guide.

- Per API i dettagli, consulta Command [StopSnomedctInferenceJob](#) Reference AWS CLI .

## AWS Config esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Config.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **delete-config-rule**

Il seguente esempio di codice mostra come utilizzare `delete-config-rule`.

## AWS CLI

Per eliminare una regola di AWS Config

Il comando seguente elimina una regola AWS Config denominata: MyConfigRule

```
aws configservice delete-config-rule --config-rule-name MyConfigRule
```

- Per API i dettagli, vedere [DeleteConfigRule](#) in AWS CLI Command Reference.

## delete-delivery-channel

Il seguente esempio di codice mostra come utilizzare `delete-delivery-channel`.

### AWS CLI

Per eliminare un canale di distribuzione

Il comando seguente elimina il canale di consegna predefinito:

```
aws configservice delete-delivery-channel --delivery-channel-name default
```

- Per API i dettagli, vedere [DeleteDeliveryChannel](#) in AWS CLI Command Reference.

## delete-evaluation-results

Il seguente esempio di codice mostra come utilizzare `delete-evaluation-results`.

### AWS CLI

Per eliminare manualmente i risultati della valutazione

Il comando seguente elimina i risultati di valutazione correnti per la regola AWS gestita s3: bucket-versioning-enabled

```
aws configservice delete-evaluation-results --config-rule-name s3-bucket-versioning-enabled
```

- Per API i dettagli, vedere [DeleteEvaluationResults](#) in AWS CLI Command Reference.

## **deliver-config-snapshot**

Il seguente esempio di codice mostra come utilizzare `deliver-config-snapshot`.

### AWS CLI

Per fornire un'istantanea della configurazione

Il comando seguente fornisce uno snapshot di configurazione al bucket Amazon S3 che appartiene al canale di distribuzione predefinito:

```
aws configservice deliver-config-snapshot --delivery-channel-name default
```

Output:

```
{
  "configSnapshotId": "d0333b00-a683-44af-921e-examplefb794"
}
```

- Per API i dettagli, consulta Command [DeliverConfigSnapshot](#) Reference AWS CLI .

## **describe-compliance-by-config-rule**

Il seguente esempio di codice mostra come utilizzare `describe-compliance-by-config-rule`.

### AWS CLI

Per ottenere informazioni sulla conformità per le regole di AWS Config

Il comando seguente restituisce informazioni sulla conformità per ogni regola di AWS Config violata da una o più risorse: AWS

```
aws configservice describe-compliance-by-config-rule --compliance-  
types NON_COMPLIANT
```

Nell'output, il valore di ogni CappedCount attributo indica quante risorse non sono conformi alla regola correlata. Ad esempio, l'output seguente indica che 3 risorse non sono conformi alla regola denominata `InstanceTypesAreT2micro`.

Output:

```
{
```



```

"ComplianceByConfigRules": [
  {
    "Compliance": {
      "ComplianceContributorCount": {
        "CappedCount": 3,
        "CapExceeded": false
      },
      "ComplianceType": "NON_COMPLIANT"
    },
    "ConfigRuleName": "InstanceTypesAreT2micro"
  },
  {
    "Compliance": {
      "ComplianceContributorCount": {
        "CappedCount": 10,
        "CapExceeded": false
      },
      "ComplianceType": "NON_COMPLIANT"
    },
    "ConfigRuleName": "RequiredTagsForVolumes"
  }
]
}

```

- Per API i dettagli, vedere [DescribeComplianceByConfigRule](#) in AWS CLI Command Reference.

## describe-compliance-by-resource

Il seguente esempio di codice mostra come utilizzare `describe-compliance-by-resource`.

### AWS CLI

Per ottenere informazioni sulla conformità delle tue AWS risorse

Il comando seguente restituisce informazioni sulla conformità per ogni EC2 istanza registrata da AWS Config e che viola una o più regole:

```

aws configservice describe-compliance-by-resource --resource-type AWS::EC2::Instance
--compliance-types NON_COMPLIANT

```

Nell'output, il valore di ogni `CappedCount` attributo indica quante regole viene violata dalla risorsa. Ad esempio, l'output seguente indica che l'istanza `i-1a2b3c4d` viola 2 regole.

**Output:**

```
{
  "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"
      }
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeComplianceByResource](#) in AWS CLI Command Reference.

**describe-config-rule-evaluation-status**

Il seguente esempio di codice mostra come utilizzare `describe-config-rule-evaluation-status`.

**AWS CLI**

Per ottenere informazioni sullo stato di una regola di AWS Config

Il comando seguente restituisce le informazioni sullo stato di una regola AWS Config denominata: `MyConfigRule`

```
aws configservice describe-config-rule-evaluation-status --config-rule-  
names MyConfigRule
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [DescribeConfigRuleEvaluationStatus](#) in AWS CLI Command Reference.

## describe-config-rules

Il seguente esempio di codice mostra come utilizzare `describe-config-rules`.

AWS CLI

Per ottenere i dettagli di una regola di AWS Config

Il comando seguente restituisce i dettagli per una regola AWS Config denominata:

`InstanceTypesAreT2micro`

```
aws configservice describe-config-rules --config-rule-names InstanceTypesAreT2micro
```

Output:

```
{  
  "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"
  }
]
}

```

- Per API i dettagli, vedere [DescribeConfigRules](#) in AWS CLI Command Reference.

## describe-configuration-recorder-status

Il seguente esempio di codice mostra come utilizzare `describe-configuration-recorder-status`.

### AWS CLI

Per ottenere informazioni sullo stato del registratore di configurazione

Il comando seguente restituisce lo stato del registratore di configurazione predefinito:

```
aws configservice describe-configuration-recorder-status
```

Output:

```
{
  "ConfigurationRecordersStatus": [
    {
      "name": "default",
      "lastStatus": "SUCCESS",
      "recording": true,
      "lastStatusChangeTime": 1452193834.344,
      "lastStartTime": 1441039997.819,
      "lastStopTime": 1441039992.835
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeConfigurationRecorderStatus](#) in AWS CLI Command Reference.

## describe-configuration-recorders

Il seguente esempio di codice mostra come utilizzare `describe-configuration-recorders`.

### AWS CLI

Per ottenere dettagli sul registratore di configurazione

Il comando seguente restituisce dettagli sul registratore di configurazione predefinito:

```
aws configservice describe-configuration-recorders
```

Output:

```
{
  "ConfigurationRecorders": [
    {
      "recordingGroup": {
        "allSupported": true,
        "resourceTypes": [],
        "includeGlobalResourceTypes": true
      },
      "roleARN": "arn:aws:iam::123456789012:role/config-ConfigRole-
A1B2C3D4E5F6",
      "name": "default"
    }
  ]
}
```

```
]
}
```

- Per API i dettagli, vedere [DescribeConfigurationRecorders](#) in AWS CLI Command Reference.

## describe-delivery-channel-status

Il seguente esempio di codice mostra come utilizzare `describe-delivery-channel-status`.

### AWS CLI

Per ottenere informazioni sullo stato del canale di consegna

Il comando seguente restituisce lo stato del canale di consegna:

```
aws configservice describe-delivery-channel-status
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeDeliveryChannelStatus](#) in AWS CLI Command Reference.

## describe-delivery-channels

Il seguente esempio di codice mostra come utilizzare `describe-delivery-channels`.

### AWS CLI

Per ottenere dettagli sul canale di consegna

Il comando seguente restituisce i dettagli sul canale di consegna:

```
aws configservice describe-delivery-channels
```

Output:

```
{
  "DeliveryChannels": [
    {
      "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
      "name": "default",
      "s3BucketName": "config-bucket-123456789012"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeDeliveryChannels](#) in AWS CLI Command Reference.

## get-compliance-details-by-config-rule

Il seguente esempio di codice mostra come utilizzare `get-compliance-details-by-config-rule`.

### AWS CLI

Per ottenere i risultati della valutazione per una regola di AWS Config

Il comando seguente restituisce i risultati della valutazione per tutte le risorse che non sono conformi a una regola AWS Config denominata: `InstanceTypesAreT2micro`

```
aws configservice get-compliance-details-by-config-rule --config-rule-
name InstanceTypesAreT2micro --compliance-types NON_COMPLIANT
```

## Output:

```
{
  "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"
    }
  ]
}
```



```
}
```

- Per API i dettagli, vedere [GetComplianceDetailsByConfigRule](#) in AWS CLI Command Reference.

## get-compliance-details-by-resource

Il seguente esempio di codice mostra come utilizzare `get-compliance-details-by-resource`.

### AWS CLI

Per ottenere i risultati della valutazione di una AWS risorsa

Il comando seguente restituisce i risultati della valutazione per ogni regola a cui l'EC2 istanza `i-1a2b3c4d` non è conforme:

```
aws configservice get-compliance-details-by-resource --resource-type AWS::EC2::Instance --resource-id i-1a2b3c4d --compliance-types NON_COMPLIANT
```

Output:

```
{
  "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"
  }
]
}

```

- Per API i dettagli, vedere [GetComplianceDetailsByResource](#) in AWS CLI Command Reference.

## get-compliance-summary-by-config-rule

Il seguente esempio di codice mostra come utilizzare `get-compliance-summary-by-config-rule`.

### AWS CLI

Per ottenere il riepilogo della conformità per le tue regole di AWS Config

Il comando seguente restituisce il numero di regole conformi e il numero di regole non conformi:

```
aws configservice get-compliance-summary-by-config-rule
```

Nell'output, il valore di ogni `CappedCount` attributo indica quante regole sono conformi o non conformi.

Output:

```

{
  "ComplianceSummary": {
    "NonCompliantResourceCount": {
      "CappedCount": 3,
      "CapExceeded": false
    },
    "ComplianceSummaryTimestamp": 1452204131.493,
    "CompliantResourceCount": {
      "CappedCount": 2,
      "CapExceeded": false
    }
  }
}

```

```
}
```

- Per i API dettagli, vedere in Command Reference. [GetComplianceSummaryByConfigRuleAWS CLI](#)

## get-compliance-summary-by-resource-type

Il seguente esempio di codice mostra come utilizzare `get-compliance-summary-by-resource-type`.

### AWS CLI

Per ottenere il riepilogo della conformità per tutti i tipi di risorse

Il comando seguente restituisce il numero di AWS risorse non conformi e il numero di risorse conformi:

```
aws configservice get-compliance-summary-by-resource-type
```

Nell'output, il valore di ogni `CappedCount` attributo indica quante risorse sono conformi o non conformi.

Output:

```
{
  "ComplianceSummariesByResourceType": [
    {
      "ComplianceSummary": {
        "NonCompliantResourceCount": {
          "CappedCount": 16,
          "CapExceeded": false
        },
        "ComplianceSummaryTimestamp": 1453237464.543,
        "CompliantResourceCount": {
          "CappedCount": 10,
          "CapExceeded": false
        }
      }
    }
  ]
}
```

Per ottenere il riepilogo della conformità per un tipo di risorsa specifico

Il comando seguente restituisce il numero di EC2 istanze non conformi e il numero di istanze conformi:

```
aws configservice get-compliance-summary-by-resource-type --resource-  
types AWS::EC2::Instance
```

Nell'output, il valore di ogni CappedCount attributo indica quante risorse sono conformi o non conformi.

Output:

```
{  
  "ComplianceSummariesByResourceType": [  
    {  
      "ResourceType": "AWS::EC2::Instance",  
      "ComplianceSummary": {  
        "NonCompliantResourceCount": {  
          "CappedCount": 3,  
          "CapExceeded": false  
        },  
        "ComplianceSummaryTimestamp": 1452204923.518,  
        "CompliantResourceCount": {  
          "CappedCount": 7,  
          "CapExceeded": false  
        }  
      }  
    }  
  ]  
}
```

- Per i API dettagli, vedere in Command Reference.  
[GetComplianceSummaryByResourceType](#) AWS CLI

## get-resource-config-history

Il seguente esempio di codice mostra come utilizzare `get-resource-config-history`.

AWS CLI

Per ottenere la cronologia di configurazione di una AWS risorsa

Il comando seguente restituisce un elenco di elementi di configurazione per un'EC2istanza con un ID di `i-1a2b3c4d`:

```
aws configservice get-resource-config-history --resource-type AWS::EC2::Instance --resource-id i-1a2b3c4d
```

- Per API i dettagli, vedere [GetResourceConfigHistory](#) in AWS CLI Command Reference.

## get-status

Il seguente esempio di codice mostra come utilizzare `get-status`.

### AWS CLI

Per ottenere lo stato di AWS Config

Il comando seguente restituisce lo stato del canale di distribuzione e del registratore di configurazione:

```
aws configservice get-status
```

Output:

```
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
```

- Per API i dettagli, vedere [GetStatus](#) in AWS CLI Command Reference.

## list-discovered-resources

Il seguente esempio di codice mostra come utilizzare `list-discovered-resources`.

## AWS CLI

Per elencare le risorse scoperte da AWS Config

Il comando seguente elenca le EC2 istanze rilevate da AWS Config:

```
aws configservice list-discovered-resources --resource-type AWS::EC2::Instance
```

Output:

```
{
  "resourceIdentifiers": [
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-1a2b3c4d"
    },
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-2a2b3c4d"
    },
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-3a2b3c4d"
    }
  ]
}
```

- Per API i dettagli, vedere [ListDiscoveredResources](#) in AWS CLI Command Reference.

## put-config-rule

Il seguente esempio di codice mostra come utilizzare `put-config-rule`.

## AWS CLI

Per aggiungere una regola AWS Config gestita

Il comando seguente fornisce il JSON codice per aggiungere una regola Config AWS gestita:

```
aws configservice put-config-rule --config-rule file://  
RequiredTagsForEC2Instances.json
```

`RequiredTagsForEC2Instances.json` è un JSON file che contiene la configurazione delle regole:

```
{
  "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\"}"
}
```

Per l'`ComplianceResourceTypes` attributo, questo JSON codice limita l'ambito alle risorse del `AWS::EC2::Instance` tipo, quindi AWS Config valuterà solo EC2 le istanze rispetto alla regola. Poiché la regola è una regola gestita, l'`Owner` attributo è impostato su e l'`SourceIdentifier` attributo è impostato sull'identificatore della regola `AWS::REQUIRED_TAGS`. Per l'`InputParameters` attributo, vengono specificate le chiavi di tag richieste dalla regola `CostCenter` e `Owner`.

Se il comando ha esito positivo, AWS Config non restituisce alcun output. Per verificare la configurazione della regola, esegui il `describe-config-rules` comando e specifica il nome della regola.

Per aggiungere una regola Config gestita dal cliente

Il comando seguente fornisce il JSON codice per aggiungere una regola Config gestita dal cliente:

```
aws configservice put-config-rule --config-rule file://InstanceTypesAreT2micro.json
```

`InstanceTypesAreT2micro.json` è un JSON file che contiene la configurazione delle regole:

```
{
  "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\"}"
}
```

Per l'`ComplianceResourceTypes` attributo, questo JSON codice limita l'ambito alle risorse del `AWS::EC2::Instance` tipo, quindi AWS Config valuterà solo EC2 le istanze rispetto alla regola. Poiché questa regola è una regola gestita dal cliente, l'`Owner` attributo è impostato su `CUSTOM_LAMBDA` e l'`SourceIdentifier` attributo è impostato sulla ARN funzione AWS Lambda. `CUSTOM_LAMBDA` L'`SourceDetails` soggetto è obbligatorio. I parametri specificati per l'`InputParameters` attributo vengono passati alla funzione AWS Lambda quando AWS Config la richiama per valutare le risorse rispetto alla regola.

Se il comando ha esito positivo, AWS Config non restituisce alcun output. Per verificare la configurazione della regola, esegui il `describe-config-rules` comando e specifica il nome della regola.

- Per API i dettagli, vedere [PutConfigRule](#) in AWS CLI Command Reference.

## put-configuration-recorder

Il seguente esempio di codice mostra come utilizzare `put-configuration-recorder`.

### AWS CLI

Esempio 1: per registrare tutte le risorse supportate



Il comando seguente crea un registratore di configurazione che tiene traccia delle modifiche a tutti i tipi di risorse supportati, inclusi i tipi di risorse globali:

```
aws configservice put-configuration-recorder \  
  --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/  
config-role \  
  --recording-group allSupported=true,includeGlobalResourceTypes=true
```

Se il comando ha esito positivo, AWS Config non restituisce alcun output. Per verificare le impostazioni del registratore di configurazione, esegui il comando `describe-configuration-recorders`

Esempio 2: Per registrare tipi specifici di risorse

Il comando seguente crea un registratore di configurazione che tiene traccia delle modifiche solo ai tipi di risorse specificati nel JSON file per l'opzione `--recording-group`:

```
aws configservice put-configuration-recorder \  
  --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/  
config-role \  
  --recording-group file://recordingGroup.json
```

`recordingGroup.json` è un JSON file che specifica i tipi di risorse che Config AWS registrerà:

```
{  
  "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"  
  ]  
}
```

Prima di poter specificare i tipi di risorse per la `resourceTypes` chiave, è necessario impostare le opzioni `allSupported` and `includeGlobalResourceTypes` su `false` o ometterle.

Se il comando ha esito positivo, AWS Config non restituisce alcun output. Per verificare le impostazioni del registratore di configurazione, esegui il comando. `describe-configuration-  
recorders`

Esempio 3: Per selezionare tutte le risorse supportate, esclusi tipi specifici di risorse

Il comando seguente crea un registratore di configurazione che tiene traccia delle modifiche a tutti i tipi di risorse supportati attuali e futuri, esclusi i tipi di risorse specificati nel JSON file per l'opzione `--recording-group`:

```
aws configservice put-configuration-recorder \  
  --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/  
config-role \  
  --recording-group file://recordingGroup.json
```

`recordingGroup.json` è un JSON file che specifica i tipi di risorse che Config AWS registrerà:

```
{  
  "allSupported": false,  
  "exclusionByResourceTypes": {  
    "resourceTypes": [  
      "AWS::Redshift::ClusterSnapshot",  
      "AWS::RDS::DBClusterSnapshot",  
      "AWS::CloudFront::StreamingDistribution"  
    ]  
  },  
  "includeGlobalResourceTypes": false,  
  "recordingStrategy": {  
    "useOnly": "EXCLUSION_BY_RESOURCE_TYPES"  
  },  
}
```

Prima di poter specificare i tipi di risorse da escludere dalla registrazione: 1) È necessario impostare le opzioni `allSupported` e `includeGlobalResourceTypes` su `false` o ometterle e 2) è necessario impostare il `useOnly` campo su `_BY_`. `RecordingStrategy EXCLUSION RESOURCE TYPES`

Se il comando ha esito positivo, AWS Config non restituisce alcun output. Per verificare le impostazioni del registratore di configurazione, esegui il comando. `describe-configuration-  
recorders`

- Per API i dettagli, vedere [PutConfigurationRecorder](#) in AWS CLI Command Reference.

## put-delivery-channel

Il seguente esempio di codice mostra come utilizzare `put-delivery-channel`.

### AWS CLI

Per creare un canale di distribuzione

Il comando seguente fornisce le impostazioni per il canale di consegna sotto forma di JSON codice:

```
aws configservice put-delivery-channel --delivery-channel file://  
deliveryChannel.json
```

Il `deliveryChannel.json` file specifica gli attributi del canale di consegna:

```
{  
  "name": "default",  
  "s3BucketName": "config-bucket-123456789012",  
  "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",  
  "configSnapshotDeliveryProperties": {  
    "deliveryFrequency": "Twelve_Hours"  
  }  
}
```

L'esempio seguente imposta i seguenti attributi:

**name**- Il nome del canale di consegna. Per impostazione predefinita, AWS Config assegna il nome `default` a un nuovo canale di consegna. Non è possibile aggiornare il nome del canale di consegna con il comando `put-delivery-channel`. Per i passaggi per modificare il nome, vedi [Ridenominazione del canale di consegna](#). **s3BucketName** - Il nome del bucket Amazon S3 a cui AWS Config fornisce istantanee di configurazione e file di cronologia della configurazione. Se si specifica un bucket che appartiene a un altro AWS account, tale bucket deve disporre di politiche che concedano le autorizzazioni di accesso a Config. AWS Per ulteriori informazioni, consulta [Autorizzazioni per il bucket Amazon S3](#).

**snsTopicARN**- L'Amazon Resource Name (ARN) dell'SNS argomento Amazon a cui AWS Config invia notifiche sulle modifiche alla configurazione. Se scegli un argomento da un altro account, l'argomento deve avere politiche che concedano le autorizzazioni di accesso a Config. AWS Per ulteriori informazioni, consulta [Autorizzazioni per l'SNS argomento Amazon](#).

`configSnapshotDeliveryProperties`- Contiene l'`deliveryFrequency` attributo, che imposta la frequenza con cui AWS Config fornisce istantanee di configurazione e la frequenza con cui richiama le valutazioni per le regole Config periodiche.

Se il comando ha esito positivo, AWS Config non restituisce alcun output. Per verificare le impostazioni del tuo canale di consegna, esegui il `describe-delivery-channels` comando.

- Per API i dettagli, consulta [PutDeliveryChannel AWS CLI Command Reference](#).

## **start-config-rules-evaluation**

Il seguente esempio di codice mostra come utilizzare `start-config-rules-evaluation`.

### AWS CLI

Per eseguire una valutazione su richiesta per le regole AWS Config

Il comando seguente avvia una valutazione per due regole AWS gestite:

```
aws configservice start-config-rules-evaluation --config-rule-names s3-bucket-  
versioning-enabled cloudtrail-enabled
```

- Per API i dettagli, vedere [StartConfigRulesEvaluation](#) in AWS CLI Command Reference.

## **start-configuration-recorder**

Il seguente esempio di codice mostra come utilizzare `start-configuration-recorder`.

### AWS CLI

Per avviare il registratore di configurazione

Il comando seguente avvia il registratore di configurazione predefinito:

```
aws configservice start-configuration-recorder --configuration-recorder-name default
```

Se il comando ha esito positivo, AWS Config non restituisce alcun output. Per verificare che AWS Config stia registrando le tue risorse, esegui il comando `get-status`.

- Per API i dettagli, vedere [StartConfigurationRecorder](#) in AWS CLI Command Reference.

## stop-configuration-recorder

Il seguente esempio di codice mostra come utilizzare `stop-configuration-recorder`.

### AWS CLI

Per arrestare il registratore di configurazione

Il comando seguente arresta il registratore di configurazione predefinito:

```
aws configservice stop-configuration-recorder --configuration-recorder-name default
```

Se il comando ha esito positivo, AWS Config non restituisce alcun output. Per verificare che AWS Config non stia registrando le tue risorse, esegui il comando `get-status`.

- Per API i dettagli, vedere [StopConfigurationRecorder](#) in AWS CLI Command Reference.

## subscribe

Il seguente esempio di codice mostra come utilizzare `subscribe`.

### AWS CLI

Per abbonarsi a AWS Config

Il comando seguente crea il canale di consegna e il registratore di configurazione predefiniti. Il comando specifica anche il bucket Amazon S3 e l'argomento SNS Amazon a AWS cui Config fornirà le informazioni di configurazione:

```
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
```

Output:

```
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"
  }
]
```

- Per API i dettagli, consulta [Subscribe](#) in Command Reference.AWS CLI

## Esempi di utilizzo di Amazon Connect AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Connect. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

### Azioni

#### **create-user**

Il seguente esempio di codice mostra come utilizzare `create-user`.

## AWS CLI

Per creare un utente

L'create-user esempio seguente aggiunge un utente con gli attributi specificati all'istanza Amazon Connect specificata.

```
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
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Add Users](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [CreateUser AWS CLI Command Reference](#).

## delete-user

Il seguente esempio di codice mostra come utilizzare delete-user.

### AWS CLI

Per eliminare un utente

L'delete-user esempio seguente elimina l'utente specificato dall'istanza Amazon Connect specificata.

```
aws connect delete-user \  

```

```
--instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
--user-id 87654321-2222-1234-1234-111234567891
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Manage Users](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [DeleteUser AWS CLI Command Reference](#).

## describe-user-hierarchy-group

Il seguente esempio di codice mostra come utilizzare `describe-user-hierarchy-group`.

### AWS CLI

Per visualizzare i dettagli di un gruppo gerarchico

L'esempio seguente mostra i dettagli per il gruppo gerarchico Amazon Connect specificato.

```
aws connect describe-user-hierarchy-group \  
--hierarchy-group-id 12345678-1111-2222-800e-aaabbb555gg \  
--instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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"  
      }  
    }  
  }  
}
```



```

    }
  }
}

```

Per ulteriori informazioni, consulta [Configurare le gerarchie degli agenti](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeUserHierarchyGroup](#)Reference.

## describe-user-hierarchy-structure

Il seguente esempio di codice mostra come utilizzare `describe-user-hierarchy-structure`.

### AWS CLI

Per visualizzare i dettagli di una struttura gerarchica

L'esempio seguente mostra i dettagli della struttura gerarchica per l'istanza Amazon Connect specificata.

```

aws connect describe-user-hierarchy-group \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Configurare le gerarchie degli agenti](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeUserHierarchyStructure](#) Reference.

## describe-user

Il seguente esempio di codice mostra come utilizzare `describe-user`.

### AWS CLI

Per visualizzare i dettagli di un utente

L'esempio seguente mostra i dettagli per l'utente Amazon Connect specificato.

```

aws connect describe-user \
  --user-id 0c245dc0-0cf5-4e37-800e-2a7481cc8a60
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e

```

Output:

```

{
  "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": {}  
}  
}
```

Per ulteriori informazioni, consulta [Manage Users](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [DescribeUser AWS CLI Command Reference](#).

## get-contact-attributes

Il seguente esempio di codice mostra come utilizzare `get-contact-attributes`.

### AWS CLI

Per recuperare gli attributi di un contatto

L'`get-contact-attributes` seguente recupera gli attributi che sono stati impostati per il contatto Amazon Connect specificato.

```
aws connect get-contact-attributes \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --initial-contact-id 12345678-1111-2222-800e-a2b3c4d5f6g7
```

Output:

```
{  
  "Attributes": {  
    "greetingPlayed": "true"  
  }  
}
```

Per ulteriori informazioni, consulta [Use Amazon Connect Contact Attributes](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [GetContactAttributes AWS CLI Command Reference](#).

## list-contact-flows

Il seguente esempio di codice mostra come utilizzare `list-contact-flows`.

### AWS CLI

Per elencare i flussi di contatti in un'istanza

L'`list-contact-flows` esempio seguente elenca i flussi di contatti nell'istanza Amazon Connect specificata.

```
aws connect list-contact-flows \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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"
    },
]
}

```

Per ulteriori informazioni, consulta [Create Amazon Connect Contact Flows](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [ListContactFlows AWS CLI Command Reference](#).

## list-hours-of-operations

Il seguente esempio di codice mostra come utilizzare `list-hours-of-operations`.

### AWS CLI

Per elencare gli orari di funzionamento di un'istanza

L'`list-hours-of-operations` esempio seguente elenca gli orari di funzionamento dell'istanza Amazon Connect specificata.

```

aws connect list-hours-of-operations \
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e

```

Output:

```

{
  "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"
    }
  ]
}

```

Per ulteriori informazioni, consulta [la sezione Impostazione degli orari di funzionamento di una coda](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [ListHoursOfOperations AWS CLI Command Reference](#).

## list-phone-numbers

Il seguente esempio di codice mostra come utilizzare `list-phone-numbers`.

### AWS CLI

Per elencare i numeri di telefono di un'istanza

L'`list-phone-numbers` esempio seguente elenca i numeri di telefono nell'istanza Amazon Connect specificata.

```
aws connect list-phone-numbers \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Configurare i numeri di telefono per il tuo contact center](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [ListPhoneNumbers AWS CLI Command Reference](#).

## list-queues

Il seguente esempio di codice mostra come utilizzare `list-queues`.

### AWS CLI

Per elencare le code in un'istanza

L'`list-queues` esempio seguente elenca le code nell'istanza Amazon Connect specificata.

```
aws connect list-queues \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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"
    },
]
}
```

Per ulteriori informazioni, consulta [Create a queue](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [ListQueues AWS CLI Command Reference](#).

## list-routing-profiles

Il seguente esempio di codice mostra come utilizzare `list-routing-profiles`.

### AWS CLI

Per elencare i profili di routing in un'istanza

L'`list-routing-profiles` esempio seguente elenca i profili di routing nell'istanza Amazon Connect specificata.

```
aws connect list-routing-profiles \
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{
  "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"
    },
  ]
}
```

Per ulteriori informazioni, consulta [Creare un profilo di routing](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [ListRoutingProfiles AWS CLI Command Reference](#).



## list-security-profiles

Il seguente esempio di codice mostra come utilizzare `list-security-profiles`.

### AWS CLI

Per elencare i profili di sicurezza in un'istanza

L'`list-security-profiles` seguente elenca i profili di sicurezza nell'istanza Amazon Connect specificata.

```
aws connect list-security-profiles \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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"
  }
]
}
```

Per ulteriori informazioni, consulta [Assegna autorizzazioni: profili di sicurezza](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta AWS CLI Command [ListSecurityProfiles](#) Reference.

## list-user-hierarchy-groups

Il seguente esempio di codice mostra come utilizzare `list-user-hierarchy-groups`.

### AWS CLI

Per elencare i gruppi gerarchici di utenti in un'istanza

L'`list-user-hierarchy-groups` seguente elenca i gruppi gerarchici di utenti nell'istanza Amazon Connect specificata.

```
aws connect list-user-hierarchy-groups \
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

Output:

```
{
  "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"
    },
  ],
}
```

Per ulteriori informazioni, consulta [Configurare le gerarchie degli agenti](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta AWS CLI Command [ListUserHierarchyGroupsReference](#).

## list-users

Il seguente esempio di codice mostra come utilizzare `list-users`.

### AWS CLI

Per elencare i gruppi gerarchici di utenti in un'istanza

L'`list-users`esempio seguente elenca gli utenti nell'istanza Amazon Connect specificata.

```
aws connect list-users \  
  --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

```
]
}
```

Per ulteriori informazioni, consulta [Add Users](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [ListUsers AWS CLI Command Reference](#).

## update-contact-attributes

Il seguente esempio di codice mostra come utilizzare `update-contact-attributes`.

### AWS CLI

Per aggiornare l'attributo di un contatto

L'esempio seguente aggiorna l'attributo `greetingPlayed` per l'utente Amazon Connect specificato.

```
aws connect update-contact-attributes \  
  --initial-contact-id 11111111-2222-3333-4444-12345678910 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --attributes greetingPlayed=false
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Use Amazon Connect Contact Attributes](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [UpdateContactAttributes AWS CLI Command Reference](#).

## update-user-hierarchy

Il seguente esempio di codice mostra come utilizzare `update-user-hierarchy`.

### AWS CLI

Per aggiornare la gerarchia di un utente

L'esempio seguente aggiorna la gerarchia degli agenti per l'utente Amazon Connect specificato.

```
aws connect update-user-hierarchy \  
  --hierarchy-group-id 12345678-a1b2-c3d4-e5f6-123456789abc \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

```
--user-id 87654321-2222-1234-1234-111234567891 \  
--instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configure Agent Settings](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [UpdateUserHierarchy AWS CLI Command Reference](#).

## update-user-identity-info

Il seguente esempio di codice mostra come utilizzare `update-user-identity-info`.

### AWS CLI

Per aggiornare le informazioni sull'identità di un utente

L'`update-user-identity-info` esempio seguente aggiorna le informazioni di identità per l'utente Amazon Connect specificato.

```
aws connect update-user-identity-info \  
  --identity-info FirstName=Mary,LastName=Major,Email=marym@example.com \  
  --user-id 87654321-2222-1234-1234-111234567891 \  
  --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configure Agent Settings](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [UpdateUserIdentityInfo AWS CLI Command Reference](#).

## update-user-phone-config

Il seguente esempio di codice mostra come utilizzare `update-user-phone-config`.

### AWS CLI

Per aggiornare la configurazione del telefono di un utente

L'`update-user-phone-config` esempio seguente aggiorna la configurazione del telefono per l'utente specificato.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configure Agent Settings](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [UpdateUserPhoneConfig AWS CLI Command Reference](#).

## update-user-routing-profile

Il seguente esempio di codice mostra come utilizzare `update-user-routing-profile`.

### AWS CLI

Per aggiornare il profilo di routing di un utente

L'esempio seguente aggiorna il profilo di routing per l'utente Amazon Connect specificato.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configure Agent Settings](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta [UpdateUserRoutingProfile AWS CLI Command Reference](#).

## update-user-security-profiles

Il seguente esempio di codice mostra come utilizzare `update-user-security-profiles`.

## AWS CLI

Per aggiornare i profili di sicurezza di un utente

L'update-user-security-profiles esempio seguente aggiorna il profilo di sicurezza per l'utente Amazon Connect specificato.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Assegna autorizzazioni: profili di sicurezza](#) nella Amazon Connect Administrator Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateUserSecurityProfiles](#) Reference.

## AWS Cost and Usage Report esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Cost and Usage Report.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **delete-report-definition**

Il seguente esempio di codice mostra come utilizzare `delete-report-definition`.

## AWS CLI

Per eliminare un rapporto AWS sui costi e sull'utilizzo

Questo esempio elimina un rapporto AWS sui costi e sull'utilizzo.

Comando:

```
aws cur --region us-east-1 delete-report-definition --report-name "ExampleReport"
```

- Per API i dettagli, vedere [DeleteReportDefinition](#) in AWS CLI Command Reference.

## describe-report-definitions

Il seguente esempio di codice mostra come utilizzare `describe-report-definitions`.

## AWS CLI

Per recuperare un elenco di report AWS sui costi e sull'utilizzo

Questo esempio descrive un elenco di report AWS sui costi e sull'utilizzo di proprietà di un account.

Comando:

```
aws cur --region us-east-1 describe-report-definitions --max-items 5
```

Output:

```
{
  "ReportDefinitions": [
    {
      "ReportName": "ExampleReport",
      "Compression": "ZIP",
      "S3Region": "us-east-1",
      "Format": "textORcsv",
      "S3Prefix": "exampleprefix",
      "S3Bucket": "example-s3-bucket",
      "TimeUnit": "DAILY",
      "AdditionalArtifacts": [
```



```

        "REDSHIFT",
        "QUICKSIGHT"
    ],
    "AdditionalSchemaElements": [
        "RESOURCES"
    ]
}
]
}

```

- Per API i dettagli, vedere [DescribeReportDefinitions](#) in AWS CLI Command Reference.

## put-report-definition

Il seguente esempio di codice mostra come utilizzare `put-report-definition`.

### AWS CLI

Per creare un rapporto AWS sui costi e sull'utilizzo

L'`put-report-definition` esempio seguente crea un rapporto giornaliero AWS sui costi e sull'utilizzo che puoi caricare su Amazon Redshift o Amazon QuickSight

```
aws cur put-report-definition --report-definition file://report-definition.json
```

Contenuto di `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"
  ]
}

```

```
}
```

- Per API i dettagli, consulta [PutReportDefinition AWS CLI Command Reference](#).

## Esempi di utilizzo del servizio Cost Explorer AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il servizio AWS Command Line Interface with Cost Explorer.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni illustrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto degli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **get-cost-and-usage**

Il seguente esempio di codice mostra come utilizzare `get-cost-and-usage`.

#### AWS CLI

Per recuperare l'utilizzo S3 di un account per il mese di settembre 2017

L'`get-cost-and-usage` seguente recupera l'utilizzo di S3 di un account per il mese di settembre 2017.

```
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
```

Contenuto di `filters.json`.

```
{
  "Dimensions": {
    "Key": "SERVICE",
    "Values": [
      "Amazon Simple Storage Service"
    ]
  }
}
```

### Output:

```
{
  "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"
        }
      }
    }
  ]
}

```

- Per API i dettagli, consulta AWS CLI Command [GetCostAndUsage](#)Reference.

## get-dimension-values

Il seguente esempio di codice mostra come utilizzare `get-dimension-values`.

### AWS CLI

Per recuperare i tag per la dimensione `SERVICE`, con il valore «Elastic»

Questo esempio recupera i tag per la dimensione `SERVICE`, con il valore «Elastic» dal 1° gennaio 2017 al 18 maggio 2017.

Comando:

```
aws ce get-dimension-values --search-string Elastic --time-  
period Start=2017-01-01,End=2017-05-18 --dimension SERVICE
```

Output:

```
{  
  "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  
}
```

- Per API i dettagli, consulta [GetDimensionValues AWS CLI Command Reference](#).

## get-reservation-coverage

Il seguente esempio di codice mostra come utilizzare `get-reservation-coverage`.

### AWS CLI

Per recuperare la copertura delle prenotazioni per le istanze EC2 `t2.nano` nella regione `us-east-1`

Questo esempio recupera la copertura delle prenotazioni per le istanze EC2 `t2.nano` nella regione `us-east-1` per luglio-settembre 2017.

Comando:

```
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"  
        ]  
      }  
    }  
  ]  
}
```

Output:

```
{  
  "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
}

```

- Per API i dettagli, vedere [GetReservationCoverage](#) in AWS CLI Command Reference.

## get-reservation-purchase-recommendation

Il seguente esempio di codice mostra come utilizzare `get-reservation-purchase-recommendation`.

### AWS CLI

Per recuperare i consigli di prenotazione per Partial Upfront EC2 RIs con un periodo di tre anni

L'`get-reservation-purchase-recommendation` esempio seguente recupera i consigli per le EC2 istanze Partial Upfront con una durata di tre anni, in base agli ultimi 60 giorni di utilizzo. 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
```

Output:

```
{  
  "Recommendations": [],  
  "Metadata": {  
    "GenerationTimestamp": "2018-08-08T15:20:57Z",  
    "RecommendationId": "00d59dde-a1ad-473f-8ff2-iexample3330b"  
  }  
}
```

- Per i API dettagli, vedere [GetReservationPurchaseRecommendation](#) in Command Reference.AWS CLI

## get-reservation-utilization

Il seguente esempio di codice mostra come utilizzare `get-reservation-utilization`.

AWS CLI

Per recuperare l'utilizzo della prenotazione per il tuo account

L'`get-reservation-utilization` esempio seguente recupera l'utilizzo RI per tutti i tipi di istanze `t2.nano` dal 01/03/2018 al 01/08/2018 per l'account.

```
aws ce get-reservation-utilization \  
--time-period Start=2018-03-01,End=2018-08-01 \  
--filter file://filters.json
```

Contenuto di `filters.json`.

```
{  
  "Dimensions": {  
    "Key": "INSTANCE_TYPE",  
    "Values": [  
      "t2.nano"  
    ]  
  }  
}
```



```
}
```

Output:

```
{
  "Total": {
    "TotalAmortizedFee": "0",
    "UtilizationPercentage": "0",
    "PurchasedHours": "0",
    "NetRISavings": "0",
    "TotalActualHours": "0",
    "AmortizedRecurringFee": "0",
    "UnusedHours": "0",
    "TotalPotentialRISavings": "0",
    "OnDemandCostOfRIHoursUsed": "0",
    "AmortizedUpfrontFee": "0"
  },
  "UtilizationsByTime": []
}
```

- API Per [GetReservationUtilization](#) AWS CLI dettagli, vedere in Command Reference.

## get-tags

Il seguente esempio di codice mostra come utilizzare `get-tags`.

### AWS CLI

Per recuperare chiavi e valori per un tag di allocazione dei costi

Questo esempio recupera tutti i tag di allocazione dei costi con la chiave «Project» e un valore che contiene "». `secretProject`

Comando:

```
aws ce get-tags --search-string secretProject --time-  
period Start=2017-01-01,End=2017-05-18 --tag-key Project
```

Output:

```
{
  "ReturnSize": 2,
```

```
"Tags": [  
  "secretProject1",  
  "secretProject2"  
],  
"TotalSize": 2  
}
```

- Per API i dettagli, vedere [GetTags](#) in AWS CLI Command Reference.

## Esempi di utilizzo di Firehose AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Firehose.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **list-delivery-streams**

Il seguente esempio di codice mostra come utilizzare `list-delivery-streams`.

AWS CLI

Per elencare i flussi di consegna disponibili

L'`list-delivery-streams` esempio seguente elenca i flussi di consegna disponibili nel tuo AWS account.

```
aws firehose list-delivery-streams
```

Output:

```
{
  "DeliveryStreamNames": [
    "my-stream"
  ],
  "HasMoreDeliveryStreams": false
}
```

Per ulteriori informazioni, consulta la sezione relativa alla [creazione di un flusso di distribuzione Amazon Kinesis Data Firehose](#) nella Guida per sviluppatori Amazon Kinesis Data Firehose.

- Per API i dettagli, consulta [ListDeliveryStreams AWS CLI Command Reference](#).

## put-record-batch

Il seguente esempio di codice mostra come utilizzare `put-record-batch`.

### AWS CLI

Per scrivere più record in uno stream

L'`put-record-batch` esempio seguente scrive tre record in uno stream. I dati sono codificati in formato Base64.

```
aws firehose put-record-batch \
  --delivery-stream-name my-stream \
  --records file://records.json
```

Contenuto di `myfile.json`.

```
[
  {"Data": "Rmlyc3QgdGhpbmc="},
  {"Data": "U2Vjb25kIHRoaW5n"},
  {"Data": "VGhpcmQgdGhpbmc="}
]
```

Output:

```
{
  "FailedPutCount": 0,
  "Encrypted": false,
  "RequestResponses": [
```

```

    {
      "RecordId": "9D20J6t2EqCTZTXwGzeSv/EVHxRoRCw89xd+o3+sXg8DhY0aWKPSmZy/
CG1RVEys1u1xbeKh6VofEYKkoeiDrcjrxhQp9iF7sUW7pujiMEQ5LzlrzCkGosxQn
+3boDnURDEaD42V7Giixp0yLJkYZcae1i7HzlCEoy9LJhMr8EjDSi40m/9Vc2uhwwuAtGE0XKpxJ2WD7ZRwtAnY1KAnv
    },
    {
      "RecordId": "jFirejqxCLlK5xjH/UNm1MVcjkTEN76I7916X9PaZ
+PVa0SXDFu1WG0qEZhxq2js7xcZ552eoeDxsuTU1MSq9nZTbVfb6cQTIXnm/GsuF37Uhg67GkmR5z9016XKJ
+/+pD1oFv7Hh9a3oUS6wYm3DcNRLTHHAimANp1PhkQvWpVLRfzbuCUkBphR2QVzhp90iHLbzGwy8/
DfH8sqWEUYASNJKS8GXP5s"
    },
    {
      "RecordId":
      "oy0amQ40o5Y2YV4vxzufdcM00w6n3EPz3tpPJGoYVVKH4APPVqNcbUgefo1stEFRg4hTLrf2k6eliHu/9+YJ5R3iie
DTBt3qBlmTj7Xq8SKVb01S7YvMTpWkMKA86f8JfmT8BMKoMb4XZS/s0kQLe+qh0sYKXW1"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Invio di dati a un flusso di distribuzione di Amazon Kinesis Data Firehose nella Amazon Kinesis Data Firehose Developer Guide](#).

- Per i API dettagli, consulta [PutRecordBatch](#) Command Reference.AWS CLI

## put-record

Il seguente esempio di codice mostra come utilizzare `put-record`.

### AWS CLI

Per scrivere un record in uno stream

L'`put-record` esempio seguente scrive i dati in uno stream. I dati sono codificati in formato Base64.

```

aws firehose put-record \
  --delivery-stream-name my-stream \
  --record '{"Data": "SGVsbG8gd29ybGQ="}'

```

Output:

```
{
```

```

    "RecordId": "RjB5K/nnoGFHqwTsZ1Nd/
    TTqvjE8V5dsyXZTQn2JXrdpMT0wssyEb6nfC8fwf1whhwnItt4mvrn+gsqeK5jB7QjuLg283+Ps4Sz/
    j1Xujv31iDhnPdaLw4B0yM9Amv7PcCuB2079RuM0NhoakbyUym1wY8yt20G8X2420wu1j1Fafhci4erAt7QhDEvpwuK8
    "Encrypted": false
  }

```

Per ulteriori informazioni, consulta [Invio di dati a un flusso di distribuzione di Amazon Kinesis Data Firehose nella Amazon Kinesis Data Firehose](#) Developer Guide.

- Per i API dettagli, consulta [PutRecord](#) Command Reference.AWS CLI

## Esempi di utilizzo di Amazon Data Lifecycle Manager AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Data Lifecycle Manager. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come chiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **create-default-role**

Il seguente esempio di codice mostra come utilizzare `create-default-role`.

#### AWS CLI

Per creare il IAM ruolo richiesto per Amazon DLM

L'dlm `create-default-role` esempio seguente crea il ruolo AWS `DataLifecycleManagerDefaultRole` predefinito per la gestione delle istantanee.

```

aws dlm create-default-role \
  --resource-type snapshot

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Ruoli di servizio predefiniti per Amazon Data Lifecycle Manager nella Amazon Elastic Compute Cloud User Guide](#).

- Per API i dettagli, consulta Command [CreateDefaultRole](#)Reference AWS CLI .

## create-lifecycle-policy

Il seguente esempio di codice mostra come utilizzare `create-lifecycle-policy`.

### AWS CLI

Per creare una politica del ciclo di vita

L'esempio seguente crea una politica del ciclo di vita che crea un'istanza giornaliera dei volumi all'ora specificata. I tag specificati vengono aggiunti alle istanze e i tag vengono inoltre copiati dal volume e aggiunti alle istanze. Se la creazione di una nuova istanza supera il numero massimo specificato, l'istanza più vecchia viene eliminata.

```
aws dlm create-lifecycle-policy \
  --description "My first policy" \
  --state ENABLED \
  --execution-role-arn arn:aws:iam::12345678910:role/  
AWSDataLifecycleManagerDefaultRole \
  --policy-details file://policyDetails.json
```

Contenuto di `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
    }
  ]
}
```

Output:

```
{
  "PolicyId": "policy-0123456789abcdef0"
}
```

- Per API i dettagli, vedere [CreateLifecyclePolicy](#) in AWS CLI Command Reference.

## delete-lifecycle-policy

Il seguente esempio di codice mostra come utilizzare `delete-lifecycle-policy`.

### AWS CLI

Per eliminare una politica del ciclo di vita

L'esempio seguente elimina la politica del ciclo di vita specificata. :

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

- Per API i dettagli, vedere [DeleteLifecyclePolicy](#) in AWS CLI Command Reference.

## get-lifecycle-policies

Il seguente esempio di codice mostra come utilizzare `get-lifecycle-policies`.

### AWS CLI

Per ottenere un riepilogo delle politiche relative al ciclo di vita

L'`get-lifecycle-policies` esempio seguente elenca tutte le politiche relative al ciclo di vita.

```
aws dlm get-lifecycle-policies
```

Output:

```
{
  "Policies": [
    {
      "PolicyId": "policy-0123456789abcdef0",
      "Description": "My first policy",
      "State": "ENABLED"
    }
  ]
}
```

- Per API i dettagli, vedere [GetLifecyclePolicies](#) in AWS CLI Command Reference.

## get-lifecycle-policy

Il seguente esempio di codice mostra come utilizzare `get-lifecycle-policy`.

### AWS CLI

Per descrivere una politica del ciclo di vita

L'`get-lifecycle-policy` esempio seguente visualizza i dettagli per la politica del ciclo di vita specificata.

```
aws dlm get-lifecycle-policy \
  --policy-id policy-0123456789abcdef0
```



## Output:

```
{
  "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
          }
        }
      ]
    }
  }
}
```

```

    }
  }
}

```

- Per API i dettagli, vedere [GetLifecyclePolicy](#) in AWS CLI Command Reference.

## update-lifecycle-policy

Il seguente esempio di codice mostra come utilizzare `update-lifecycle-policy`.

### AWS CLI

Esempio 1: abilitare una politica del ciclo di vita

L'`update-lifecycle-policy` esempio seguente abilita la politica del ciclo di vita specificata.

```

aws dlm update-lifecycle-policy \
  --policy-id policy-0123456789abcdef0 \
  --state ENABLED

```

Esempio 2: disabilitare una politica del ciclo di vita

L'`update-lifecycle-policy` esempio seguente disabilita la politica del ciclo di vita specificata.

```

aws dlm update-lifecycle-policy \
  --policy-id policy-0123456789abcdef0 \
  --state DISABLED

```

Esempio 3: per aggiornare i dettagli della politica del ciclo di vita

L'`update-lifecycle-policy` esempio seguente aggiorna i tag di destinazione per la politica del ciclo di vita specificata.

```

aws dlm update-lifecycle-policy \
  --policy-id policy-0123456789abcdef0 \
  --policy-details file://policyDetails.json

```

Contenuto di `policyDetails.json`. Gli altri dettagli non citati in questo file non vengono modificati dal comando.

```

{

```

```
    "TargetTags": [
      {
        "Key": "costCenter",
        "Value": "120"
      },
      {
        "Key": "project",
        "Value": "lima"
      }
    ]
  }
```

- Per API i dettagli, vedere [UpdateLifecyclePolicy](#) in AWS CLI Command Reference.

## AWS Data Pipeline esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Data Pipeline.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **activate-pipeline**

Il seguente esempio di codice mostra come utilizzare `activate-pipeline`.

#### AWS CLI

Per attivare una pipeline

Questo esempio attiva la pipeline specificata:

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Per attivare la pipeline in una data e un'ora specifiche, utilizzate il seguente comando:

```
aws datapipeline activate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE --start-timestamp 2015-04-07T00:00:00Z
```

- Per API i dettagli, vedere [ActivatePipeline](#) in AWS CLI Command Reference.

## add-tags

Il seguente esempio di codice mostra come utilizzare `add-tags`.

### AWS CLI

Per aggiungere un tag a una pipeline

Questo esempio aggiunge il tag specificato alla pipeline specificata:

```
aws datapipeline add-tags --pipeline-id df-00627471S0VYZEXAMPLE --tags key=environment,value=production key=owner,value=sales
```

Per visualizzare i tag, utilizzate il comando `describe-pipelines`. Ad esempio, i tag aggiunti nel comando `example` vengono visualizzati come segue nell'output di `describe-pipelines`:

```
{
  ...
  "tags": [
    {
      "value": "production",
      "key": "environment"
    },
    {
      "value": "sales",
      "key": "owner"
    }
  ]
  ...
}
```

- Per i API dettagli, vedere [AddTags](#) in Command Reference.AWS CLI

## create-pipeline

Il seguente esempio di codice mostra come utilizzare `create-pipeline`.

### AWS CLI

Per creare una pipeline

Questo esempio crea una pipeline:

```
aws datapipeline create-pipeline --name my-pipeline --unique-id my-pipeline-token
```

Di seguito è riportato un output di esempio:

```
{  
  "pipelineId": "df-00627471S0VYZEXAMPLE"  
}
```

- Per API i dettagli, vedere [CreatePipeline](#) in AWS CLI Command Reference.

## deactivate-pipeline

Il seguente esempio di codice mostra come utilizzare `deactivate-pipeline`.

### AWS CLI

Per disattivare una pipeline

Questo esempio disattiva la pipeline specificata:

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

Per disattivare la pipeline solo al termine di tutte le attività in esecuzione, utilizzate il comando seguente:

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471S0VYZEXAMPLE --no-cancel-active
```

- Per API i dettagli, vedere [DeactivatePipeline](#) in AWS CLI Command Reference.

## delete-pipeline

Il seguente esempio di codice mostra come utilizzare `delete-pipeline`.

### AWS CLI

Per eliminare una pipeline

Questo esempio elimina la pipeline specificata:

```
aws datapipeline delete-pipeline --pipeline-id df-00627471S0VYZEXAMPLE
```

- Per API i dettagli, vedere [DeletePipeline](#) in AWS CLI Command Reference.

## describe-pipelines

Il seguente esempio di codice mostra come utilizzare `describe-pipelines`.

### AWS CLI

Per descrivere le tue pipeline

Questo esempio descrive la pipeline specificata:

```
aws datapipeline describe-pipelines --pipeline-ids df-00627471S0VYZEXAMPLE
```

Di seguito è riportato un output di esempio:

```
{
  "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": []
  }
]
}

```

- Per API i dettagli, vedere [DescribePipelines](#) in AWS CLI Command Reference.

## get-pipeline-definition

Il seguente esempio di codice mostra come utilizzare `get-pipeline-definition`.

### AWS CLI

Per ottenere una definizione di pipeline

Questo esempio ottiene la definizione della pipeline per la pipeline specificata:

```
aws datapipeline get-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE
```

Di seguito è riportato un output di esempio:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [GetPipelineDefinition](#) in AWS CLI Command Reference.

## list-pipelines

Il seguente esempio di codice mostra come utilizzare `list-pipelines`.

### AWS CLI

Per elencare le tue pipeline

Questo esempio elenca le tue pipeline:

```
aws datapipeline list-pipelines
```

Di seguito è riportato un output di esempio:

```
{
  "pipelineIdList": [
    {
      "id": "df-00627471S0VYZEXAMPLE",
      "name": "my-pipeline"
    },
    {
      "id": "df-09028963KNVMREXAMPLE",
      "name": "ImportDDB"
    },
    {
      "id": "df-0870198233ZYVEXAMPLE",
      "name": "CrossRegionDDB"
    },
    {
      "id": "df-00189603TB4MZEXAMPLE",
      "name": "CopyRedshift"
    }
  ]
}
```

```
}

```

- Per API i dettagli, vedere [ListPipelines](#) in AWS CLI Command Reference.

## list-runs

Il seguente esempio di codice mostra come utilizzare `list-runs`.

### AWS CLI

Esempio 1: per elencare le esecuzioni della pipeline

L'`list-runs` seguente elenca le esecuzioni per la pipeline specificata.

```
aws datapipeline list-runs --pipeline-id df-00627471S0VYZEXAMPLE
```

Output:

Name	Scheduled Start	Status	Ended	ID
	Started			
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		
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		

Esempio 2: Per elencare le corse della pipeline tra le date specificate

L'`list-runs` seguente utilizza `--start-interval` per specificare le date da includere nell'output.

```
aws datapipeline list-runs --pipeline-id df-01434553B58A2SHZUK05 --start-interval 2017-10-07T00:00:00,2017-10-08T00:00:00
```

- Per API i dettagli, vedere [ListRuns](#) in AWS CLI Command Reference.

## put-pipeline-definition

Il seguente esempio di codice mostra come utilizzare `put-pipeline-definition`.

### AWS CLI

Per caricare una definizione di pipeline

Questo esempio carica la definizione della tubazione specificata nella tubazione specificata:

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471S0VYZEXAMPLE --  
pipeline-definition file://my-pipeline-definition.json
```

Di seguito è riportato un output di esempio:

```
{  
  "validationErrors": [],  
  "errored": false,  
  "validationWarnings": []  
}
```

- Per API i dettagli, vedete [PutPipelineDefinition](#) in AWS CLI Command Reference.

## remove-tags

Il seguente esempio di codice mostra come utilizzare `remove-tags`.

### AWS CLI

Per rimuovere un tag da una pipeline

Questo esempio rimuove il tag specificato dalla pipeline specificata:

```
aws datapipeline remove-tags --pipeline-id df-00627471S0VYZEXAMPLE --tag-  
keys environment
```

- Per API i dettagli, vedere [RemoveTags](#) in AWS CLI Command Reference.

## DataSync esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with DataSync.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **update-location-azure-blob**

Il seguente esempio di codice mostra come utilizzare `update-location-azure-blob`.

AWS CLI

Per aggiornare la località di trasferimento con un nuovo agente

L'`update-location-object-storage` esempio seguente aggiorna la DataSync posizione per Microsoft Azure Blob Storage con un nuovo agente.

```
aws datasync update-location-azure-blob \  
  --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-  
abcdef01234567890 \  
  --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/  
agent-1234567890abcdef0 \  
  --sas-configuration '{ \  
    "Token": "sas-token-for-azure-blob-storage-access" \  
  }'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Sostituzione dell'agente nella Guida](#) per l'AWS DataSync utente.

- Per API i dettagli, vedere [UpdateLocationAzureBlob](#) in AWS CLI Command Reference.

## update-location-hdfs

Il seguente esempio di codice mostra come utilizzare `update-location-hdfs`.

### AWS CLI

Per aggiornare la località di trasferimento con un nuovo agente

L'update-location-hdfs esempio seguente aggiorna la tua DataSync HDFS posizione con un nuovo agente. Le `--kerberos-krb5-conf` opzioni `--kerberos-keytab` and sono necessarie solo se il HDFS cluster utilizza l'autenticazione Kerberos.

```
aws datasync update-location-hdfs \  
  --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-  
abcdef01234567890 \  
  --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/  
agent-1234567890abcdef0 \  
  --kerberos-keytab file://hdfs.keytab \  
  --kerberos-krb5-conf file://krb5.conf
```

Contenuto di `hdfs.keytab`.

```
N/A. The content of this file is encrypted and not human readable.
```

Contenuto di `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

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Sostituzione dell'agente nella Guida](#) per l'AWS DataSync utente.

- Per API i dettagli, vedere [UpdateLocationHdfs](#) in AWS CLI Command Reference.

## update-location-nfs

Il seguente esempio di codice mostra come utilizzare `update-location-nfs`.

### AWS CLI

Per aggiornare la località di trasferimento con un nuovo agente

L'`update-location-nfs` esempio seguente aggiorna la tua DataSync NFS posizione con un nuovo agente.

```

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

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Sostituzione dell'agente](#) nella Guida AWS DataSync per l'utente.

- Per API i dettagli, vedere [UpdateLocationNfs](#) in AWS CLI Command Reference.

## update-location-object-storage

Il seguente esempio di codice mostra come utilizzare `update-location-object-storage`.

### AWS CLI

Per aggiornare la località di trasferimento con un nuovo agente

L'esempio seguente aggiorna la posizione di archiviazione DataSync degli oggetti con un nuovo agente.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [consultate Sostituzione dell'agente](#) nella Guida AWS DataSync per l'utente.

- Per API i dettagli, vedere [UpdateLocationObjectStorage](#) in AWS CLI Command Reference.

## update-location-smb

Il seguente esempio di codice mostra come utilizzare `update-location-smb`.

### AWS CLI

Per aggiornare la località di trasferimento con un nuovo agente

L'esempio seguente aggiorna la tua DataSync SMB posizione con un nuovo agente.

```
aws datasync update-location-smb \  
  --location-arn arn:aws:datasync:us-west-2:123456789012:location/Loc-  
abcdef01234567890 \  
  --secret-key secret-key-for-object-storage
```



```
--agent-arns arn:aws:datasync:us-west-2:123456789012:agent/  
agent-1234567890abcdef0 \  
--password smb-file-server-password
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Sostituzione dell'agente](#) nella Guida AWS DataSync per l'utente.

- Per API i dettagli, vedere [UpdateLocationSmb](#) in AWS CLI Command Reference.

## DAX esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with DAX.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-cluster**

Il seguente esempio di codice mostra come utilizzare `create-cluster`.

AWS CLI

Per creare un DAX cluster

L'`create-cluster` esempio seguente crea un DAX cluster con le impostazioni specificate.

```
aws dax create-cluster \  
--cluster-name daxcluster \  
--agent-arns arn:aws:datasync:us-west-2:123456789012:agent/  
agent-1234567890abcdef0 \  
--password smb-file-server-password
```

```
--node-type dax.r4.large \  
--replication-factor 3 \  
--iam-role-arn roleARN \  
--sse-specification Enabled=true
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta la [Fase 3: Creare un DAX cluster](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateCluster](#) Reference.

## create-parameter-group

Il seguente esempio di codice mostra come utilizzare `create-parameter-group`.

### AWS CLI

Per creare un gruppo di parametri

Il seguente esempio `create-parameter-group` crea un gruppo di parametri con le impostazioni specificate.

```
aws dax create-parameter-group \  
  --parameter-group-name daxparametergroup \  
  --description "A new parameter group"
```

Output:

```
{  
  "ParameterGroup": {  
    "ParameterGroupName": "daxparametergroup",  
    "Description": "A new parameter group"  
  }  
}
```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [CreateParameterGroup](#) Reference AWS CLI .

## create-subnet-group

Il seguente esempio di codice mostra come utilizzare `create-subnet-group`.

### AWS CLI

Per creare un gruppo di DAX sottoreti

L'esempio seguente `create-subnet-group` crea un gruppo di sottoreti con le impostazioni specificate.

```
aws dax create-subnet-group \  
  --parameter-group-name daxparametergroup \  
  --description "A new parameter group"
```

```
--subnet-group-name daxSubnetGroup \  
--subnet-ids subnet-11111111 subnet-22222222
```

Output:

```
{  
  "SubnetGroup": {  
    "SubnetGroupName": "daxSubnetGroup",  
    "VpcId": "vpc-05a1fa8e00c325226",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-11111111",  
        "SubnetAvailabilityZone": "us-west-2b"  
      },  
      {  
        "SubnetIdentifier": "subnet-22222222",  
        "SubnetAvailabilityZone": "us-west-2c"  
      }  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta la [Fase 2: Creare un gruppo di sottoreti](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [CreateSubnetGroup](#)Reference AWS CLI .

## decrease-replication-factor

Il seguente esempio di codice mostra come utilizzare `decrease-replication-factor`.

AWS CLI

Per rimuovere uno o più nodi dal cluster

L'`decrease-replication-factor`esempio seguente riduce a uno il numero di nodi nel DAX cluster specificato.

```
aws dax decrease-replication-factor \  
  --cluster-name daxcluster \  
  --new-replication-factor 1
```

## Output:

```
{
  "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"
  }
}
}

```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [DecreaseReplicationFactor](#) Reference AWS CLI .

## delete-cluster

Il seguente esempio di codice mostra come utilizzare `delete-cluster`.

### AWS CLI

Per eliminare un DAX cluster

L'`delete-cluster` esempio seguente elimina il DAX cluster specificato.

```
aws dax delete-cluster \  
  --cluster-name daxcluster
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [DeleteCluster](#) Reference AWS CLI .

## delete-parameter-group

Il seguente esempio di codice mostra come utilizzare `delete-parameter-group`.

### AWS CLI

Per eliminare un gruppo di parametri

L'`delete-parameter-group`esempio seguente elimina il gruppo di DAX parametri specificato.

```
aws dax delete-parameter-group \  
  --parameter-group-name daxparametergroup
```

Output:

```
{  
  "DeletionMessage": "Parameter group daxparametergroup has been deleted."  
}
```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [DeleteParameterGroup](#)Reference AWS CLI .

## delete-subnet-group

Il seguente esempio di codice mostra come utilizzare `delete-subnet-group`.

### AWS CLI

Per eliminare un gruppo di sottoreti

L'`delete-subnet-group`esempio seguente elimina il gruppo di DAX sottoreti specificato.

```
aws dax delete-subnet-group \  
  --subnet-group-name daxSubnetGroup
```

Output:

```
{  
  "DeletionMessage": "Subnet group daxSubnetGroup has been deleted."  
}
```



Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [DeleteSubnetGroup](#)Reference AWS CLI .

## describe-clusters

Il seguente esempio di codice mostra come utilizzare `describe-clusters`.

### AWS CLI

Per restituire informazioni su tutti i cluster assegnati DAX

L'`describe-clusters`esempio seguente visualizza i dettagli su tutti i cluster a cui è stato assegnato il provisioningDAX.

```
aws dax describe-clusters
```

Output:

```
{
  "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"
}
}
]
}

```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [DescribeClusters](#)Reference AWS CLI .

## **describe-default-parameters**

Il seguente esempio di codice mostra come utilizzare `describe-default-parameters`.

### AWS CLI

Per restituire le informazioni sui parametri di sistema predefiniti per DAX

L'`describe-default-parameters`esempio seguente visualizza le informazioni sui parametri di sistema predefiniti perDAX.

**aws dax describe-default-parameters**

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [DescribeDefaultParameters](#)Reference AWS CLI .

**describe-events**

Il seguente esempio di codice mostra come utilizzare `describe-events`.

## AWS CLI

Per restituire tutti gli eventi relativi ai DAX cluster e ai gruppi di parametri

L'`describe-events` seguente visualizza i dettagli degli eventi correlati ai DAX cluster e ai gruppi di parametri.

```
aws dax describe-events
```

Output:

```
{
  "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
    }
  ]
}
```

```
]
}
```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [DescribeEvents](#) Reference AWS CLI .

## **describe-parameter-groups**

Il seguente esempio di codice mostra come utilizzare `describe-parameter-groups`.

AWS CLI

Per descrivere i gruppi di parametri definiti in DAX

L'`describe-parameter-groups` esempio seguente recupera i dettagli sui gruppi di parametri definiti in DAX.

```
aws dax describe-parameter-groups
```

Output:

```
{
  "ParameterGroups": [
    {
      "ParameterGroupName": "default.dax1.0",
      "Description": "Default parameter group for dax1.0"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [DescribeParameterGroups](#) Reference AWS CLI .

## **describe-parameters**

Il seguente esempio di codice mostra come utilizzare `describe-parameters`.

## AWS CLI

Per descrivere i parametri definiti in un gruppo di DAX parametri

L'`describe-parameters` esempio seguente recupera i dettagli sui parametri definiti nel gruppo di DAX parametri specificato.

```
aws dax describe-parameters \  
  --parameter-group-name default.dax1.0
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [DescribeParametersReference](#) AWS CLI .

## describe-subnet-groups

Il seguente esempio di codice mostra come utilizzare `describe-subnet-groups`.

### AWS CLI

Per descrivere i gruppi di sottoreti definiti in DAX

L'`describe-subnet-groups` seguente recupera i dettagli per i gruppi di sottoreti definiti in DAX

```
aws dax describe-subnet-groups
```

Output:

```
{
  "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"
        }
      ]
    }
  ]
}
```

```
    ]
  }
]
}
```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [DescribeSubnetGroups](#) Reference AWS CLI .

## increase-replication-factor

Il seguente esempio di codice mostra come utilizzare `increase-replication-factor`.

### AWS CLI

Per aumentare il fattore di replica per un cluster DAX

L'`increase-replication-factor` seguente aumenta il fattore di replica del DAX cluster specificato a 3.

```
aws dax increase-replication-factor \
  --cluster-name daxcluster \
  --new-replication-factor 3
```

Output:

```
{
  "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.eyJ3o9d.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"
}
}
}

```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [IncreaseReplicationFactor](#)Reference AWS CLI .

## list-tags

Il seguente esempio di codice mostra come utilizzare `list-tags`.

### AWS CLI

Per elencare i tag su una DAX risorsa

L'`list-tags`esempio seguente elenca le chiavi e i valori dei tag associati al DAX cluster specificato.

```
aws dax list-tags \  
  --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster
```

Output:

```
{  
  "Tags": [  
    {  
      "Key": "ClusterUsage",  
      "Value": "prod"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [ListTags](#)Reference AWS CLI .

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare una DAX risorsa

L'`tag-resource`esempio seguente allega il nome della chiave del tag specificato e il valore associato al DAX cluster specificato per descrivere l'utilizzo del cluster.

```
aws dax tag-resource \  
  --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \  
  --tag-key ClusterUsage --tag-value prod
```

```
--tags="Key=ClusterUsage,Value=prod"
```

Output:

```
{
  "Tags": [
    {
      "Key": "ClusterUsage",
      "Value": "prod"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [TagResource](#)Reference AWS CLI .

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

AWS CLI

Per rimuovere i tag da una DAX risorsa

L'`untag-resource` esempio seguente rimuove il tag con il nome di chiave specificato da un DAX cluster.

```
aws dax untag-resource \
  --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \
  --tag-keys="ClusterUsage"
```

Output:

```
{
  "Tags": []
}
```

Per ulteriori informazioni, consulta [Managing DAX Clusters](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [UntagResource](#)Reference AWS CLI .

## Esempi di Detective che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Detective.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **accept-invitation**

Il seguente esempio di codice mostra come utilizzare `accept-invitation`.

AWS CLI

Accettare un invito a diventare un account membro in un grafico comportamentale

L'`accept-invitation` esempio seguente accetta un invito a diventare un account membro nel grafico comportamentale `arn:aws:detective:us-east-1:111122223333:graph:123412341234`.

```
aws detective accept-invitation \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Rispondere a un invito con grafico comportamentale](#) nella Amazon Detective Administration Guide.

- Per API i dettagli, consulta [AcceptInvitation AWS CLI](#)Command Reference.

## create-graph

Il seguente esempio di codice mostra come utilizzare `create-graph`.

### AWS CLI

Per abilitare Amazon Detective e creare un nuovo grafico comportamentale

L'esempio seguente abilita Detective per l'AWS account che esegue il comando nella regione in cui viene eseguito il comando. Viene creato un nuovo grafico comportamentale con quell'account come account amministratore. Il comando assegna anche il valore Finance al tag Department.

```
aws detective create-graph \  
  --tags '{"Department": "Finance"}'
```

Output:

```
{  
  "GraphArn": "arn:aws:detective:us-  
east-1:111122223333:graph:027c7c4610ea4aacaf0b883093cab899"  
}
```

Per ulteriori informazioni, consulta [Enabling Amazon Detective](#) nella Amazon Detective Administration Guide.

- Per API i dettagli, consulta [CreateGraph AWS CLI Command Reference](#).

## create-members

Il seguente esempio di codice mostra come utilizzare `create-members`.

### AWS CLI

Per invitare gli account dei membri a visualizzare un grafico comportamentale

L'esempio seguente invita due AWS account a diventare account membro nel grafico di comportamento `arn:aws:detective:us-east-1:111122223333:graph:123412341234`. Per ogni account, la richiesta fornisce l'ID dell'account e l'indirizzo e-mail dell'AWS utente root dell'account. La richiesta include un messaggio personalizzato da inserire nell'e-mail di invito.

```
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."
```

Output:

```
{
  "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": [ ]
}
```

Per ulteriori informazioni, consulta [Invitare gli account dei membri a un grafico comportamento](https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html) nella Amazon Detective Administration Guide.

Per invitare account membri senza inviare e-mail di invito

L'create-membersesempio seguente invita due AWS account a diventare account membro nel grafico di comportamento `arn:aws:detective:us-east-1:111122223333:graph:123412341234`. Per ogni account, la richiesta fornisce l'ID dell'account e l'indirizzo e-mail dell' AWS utente root dell'account. Gli account dei membri non ricevono e-mail di invito.

```
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
```

Output:

```
{
  "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": [ ]
}
```

Per ulteriori informazioni, consulta [Invitare gli account dei membri a un grafico comportamento](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`> nella Amazon Detective Administration Guide.

- Per i API dettagli, consulta Command Reference. [CreateMembers](#)AWS CLI

## delete-graph

Il seguente esempio di codice mostra come utilizzare `delete-graph`.

### AWS CLI

Per disabilitare Detective ed eliminare il grafico del comportamento

L'`delete-graph`esempio seguente disattiva Detective ed elimina il grafico del comportamento specificato.

```
aws detective delete-graph \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Disabling Amazon Detective](#) nella Amazon Detective Administration Guide.

- Per API i dettagli, consulta [DeleteGraph AWS CLI](#)Command Reference.

## delete-members

Il seguente esempio di codice mostra come utilizzare `delete-members`.

### AWS CLI

Per rimuovere gli account dei membri da un grafico comportamentale

L'`delete-members`esempio seguente rimuove due account membri dal grafico comportamentale `arn:aws:detective:us-east-1:111122223333:graph:123412341234`. Per identificare gli AWS IDs account, la richiesta fornisce l'account.

```
aws detective delete-members \  
  --account-ids 444455556666 123456789012 \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```



## Output:

```
{
  "AccountIds": [ "444455556666", "123456789012" ],
  "UnprocessedAccounts": [ ]
}
```

Per ulteriori informazioni, consulta [Rimuovere gli account dei membri da un grafico comportamento](https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-remove-member-accounts.html) < <https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-remove-member-accounts.html> > nella Amazon Detective Administration Guide.

- Per i API dettagli, consulta [DeleteMembers](#) Command Reference.AWS CLI

## disassociate-membership

Il seguente esempio di codice mostra come utilizzare `disassociate-membership`.

### AWS CLI

Rinunciare all'appartenenza a un grafico comportamentale

Il seguente esempio di `disassociate-membership` rimuove l' AWS account che esegue il comando dal grafico del comportamento `arn:aws:detective:us-east-1:111122223333:graph:123412341234`.

```
aws detective disassociate-membership \
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Per ulteriori informazioni, consulta [Rimuovere l'account da un grafico comportamento](https://docs.aws.amazon.com/detective/latest/adminguide/member-remove-self-from-graph.html) < <https://docs.aws.amazon.com/detective/latest/adminguide/member-remove-self-from-graph.html> > nella Amazon Detective Administration Guide.

- Per i API dettagli, consulta [DisassociateMembership](#) Command Reference.AWS CLI

## get-members

Il seguente esempio di codice mostra come utilizzare `get-members`.

### AWS CLI

Per recuperare informazioni sugli account dei membri del grafico comportamentale selezionati

L'get-members esempio seguente recupera informazioni su due account membri nel grafico comportamentale arn:aws:detective:us-east-1:111122223333:graph:123412341234. Per i due account AWS, IDs la richiesta fornisce l'account.

```
aws detective get-members \  
  --account-ids 444455556666 123456789012 \  
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Output:

```
{  
  "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": [ ]  
}
```

Per ulteriori informazioni, consulta Visualizzazione dell'elenco degli account in un grafico comportamento < <https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-view-accounts.html> > nella Amazon Detective Administration Guide.

- Per i API dettagli, consulta [GetMembers](#) Command Reference.AWS CLI

## list-graphs

Il seguente esempio di codice mostra come utilizzare `list-graphs`.

### AWS CLI

Per visualizzare un elenco di grafici comportamentali di cui l'account è l'amministratore

L'`list-graphs`esempio seguente recupera i grafici comportamentali di cui l'account chiamante è l'amministratore nella regione corrente.

```
aws detective list-graphs
```

Output:

```
{
  "GraphList": [
    {
      "Arn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
      "CreatedTime": 1579736111000
    }
  ]
}
```

- Per API i dettagli, vedere [ListGraphs](#) in AWS CLI Command Reference.

## list-invitations

Il seguente esempio di codice mostra come utilizzare `list-invitations`.

### AWS CLI

Per visualizzare un elenco di grafici comportamentali di cui un account è membro o a cui è invitato

L'`list-invitations`esempio seguente recupera i grafici comportamentali a cui l'account chiamante è stato invitato. I risultati includono solo gli inviti aperti e accettati. Non includono gli inviti rifiutati o le iscrizioni rimosse.

```
aws detective list-invitations
```

Output:

```
{
  "Invitations": [
    {
      "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
    }
  ]
}
```

Per ulteriori informazioni, consulta Visualizzazione dell'elenco degli inviti relativi al grafico del comportamento < <https://docs.aws.amazon.com/detective/latest/adminguide/member-view-graph-invitations.html> > nella Amazon Detective Administration Guide.

- Per i API dettagli, consulta Command Reference. [ListInvitations](#) AWS CLI

## list-members

Il seguente esempio di codice mostra come utilizzare `list-members`.

### AWS CLI

Per elencare gli account dei membri in un grafico comportamentale

L'`list-members` esempio seguente recupera gli account dei membri invitati e abilitati per il grafico `arn:aws:detective:us-east-1:111122223333:graph:123412341234` del comportamento. I risultati non includono gli account dei membri che sono stati rimossi.

```
aws detective list-members \
  --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Output:

```
{
  "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
  }
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione dell'elenco degli account in un grafico comportamentale](#) nell'Amazon Detective Administration Guide.

- Per API i dettagli, consulta [ListMembers AWS CLI](#) Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per recuperare i tag assegnati a un grafico di comportamento

L'`list-tags-for-resource` esempio seguente restituisce i tag assegnati al grafico di comportamento specificato.

```
aws detective list-tags-for-resource \  
--resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Output:

```
{  
  "Tags": {  
    "Department" : "Finance"  
  }  
}
```

Per ulteriori informazioni, consulta [Managing tags for a behavior graph](#) in Amazon Detective Administration Guide.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## reject-invitation

Il seguente esempio di codice mostra come utilizzare `reject-invitation`.

AWS CLI

Rifiutare un invito a diventare un account membro in un grafico comportamentale

L'`reject-invitation` seguente rifiuta un invito a diventare un account membro nel grafico comportamentale `arn:aws:detective:us-east-1:111122223333:graph:123412341234`.

```
aws detective reject-invitation \  
--graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Rispondere a un invito con grafico comportamento](https://docs.aws.amazon.com/detective/latest/adminguide/member-invitation-response.html) < <https://docs.aws.amazon.com/detective/latest/adminguide/member-invitation-response.html> > nella Amazon Detective Administration Guide.

- Per i API AWS CLI dettagli, [RejectInvitation](#) consulta Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

## AWS CLI

Per assegnare un tag a una risorsa

L'`tag-resource` esempio seguente assegna un valore per il tag Department al grafico di comportamento specificato.

```
aws detective tag-resource \  
  --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \  
  --tags '{"Department": "Finance"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing tags for a behavior graph](#) in Amazon Detective Administration Guide.

- Per API i dettagli, consulta [TagResource AWS CLI Command Reference](#).

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

## AWS CLI

Per rimuovere un valore di tag da una risorsa

L'`untag-resource` esempio seguente rimuove il tag Department dal grafico di comportamento specificato.

```
aws detective untag-resource \  
  --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \  
  --tag-keys "Department"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing tags for a behavior graph](#) in Amazon Detective Administration Guide.

- Per API i dettagli, consulta [UntagResource AWS CLI Command Reference](#).

## Esempi di Device Farm utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Device Farm.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-device-pool**

Il seguente esempio di codice mostra come utilizzare `create-device-pool`.

AWS CLI

Per creare un pool di dispositivi

Il comando seguente crea un pool di dispositivi Android per un progetto:

```
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"
```

È possibile ottenere il progetto ARN dall'output di `create-project` o `list-projects`. Il file `device-pool-rules.json` è un JSON documento nella cartella corrente che specifica la piattaforma del dispositivo:

```
[
  {
    "attribute": "PLATFORM",
    "operator": "EQUALS",
    "value": "\"ANDROID\""
  }
]
```



```
]
```

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [CreateDevicePool](#) in AWS CLI Command Reference.

## create-project

Il seguente esempio di codice mostra come utilizzare `create-project`.

### AWS CLI

Per creare un progetto

Il comando seguente crea un nuovo progetto denominato `my-project`:

```
aws devicefarm create-project --name my-project
```

Output:

```
{
  "project": {
    "name": "myproject",
    "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506",
```

```

    "created": 1503612890.057
  }
}

```

- Per API i dettagli, vedere [CreateProject](#) in AWS CLI Command Reference.

## create-upload

Il seguente esempio di codice mostra come utilizzare `create-upload`.

### AWS CLI

Per creare un caricamento

Il comando seguente crea un caricamento per un'app Android:

```

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

```

È possibile ottenere il progetto ARN dall'output di `create-project` o `list-projects`.

Output:

```

{
  "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%2Faws4_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"
  }
}

```

```
}
```

Usa l'output URL registrato per caricare un file su Device Farm:

```
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=AKIAEXAMPLEBUMBC3GA%2F20170824%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Signature=05050370c38894ef5bd09f5d009f36fc8f96fa4bb04e1bba9aca71b8dbe49a0f"
```

- Per API i dettagli, vedere [CreateUpload](#) in AWS CLI Command Reference.

## get-upload

Il seguente esempio di codice mostra come utilizzare `get-upload`.

### AWS CLI

Per visualizzare un caricamento

Il comando seguente recupera le informazioni su un caricamento:

```
aws devicefarm get-upload --arn "arn:aws:devicefarm:us-west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-ae9e-4087-09e6-f4cea3599514"
```

È possibile ottenere il caricamento ARN dall'output di `create-upload`.

Output:

```
{
  "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\":
\\\"ccom.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\\\"}"
    }
  }
}

```

- Per API i dettagli, vedere [GetUpload](#) in AWS CLI Command Reference.

## list-projects

Il seguente esempio di codice mostra come utilizzare `list-projects`.

### AWS CLI

Per elencare i progetti

Di seguito viene recuperato un elenco di progetti:

```
aws devicefarm list-projects
```

Output:

```

{
  "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
    }
  ]
}

```

- Per API i dettagli, vedere [ListProjects](#) in AWS CLI Command Reference.

## AWS Direct Connect esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Direct Connect.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **accept-direct-connect-gateway-association-proposal**

Il seguente esempio di codice mostra come utilizzare `accept-direct-connect-gateway-association-proposal`.

#### AWS CLI

Accettare una proposta di associazione gateway

Quando segue `accept-direct-connect-gateway-association-proposal` accetta la proposta specificata.

```
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"
    }
  ]
}
}

```

Per ulteriori informazioni, vedere [Accettazione o rifiuto di una proposta di associazione Transit Gateway](#) nella Guida per l'utente di AWS Direct Connect.

- Per API i dettagli, vedere [AcceptDirectConnectGatewayAssociationProposal](#) in AWS CLI Command Reference.

## allocate-connection-on-interconnect

Il seguente esempio di codice mostra come utilizzare `allocate-connection-on-interconnect`.

### AWS CLI

Per creare una connessione ospitata su un'interconnessione

Il `allocate-connection-on-interconnect` comando seguente crea una connessione ospitata su un'interconnessione:

```

aws directconnect allocate-connection-on-interconnect --bandwidth 500Mbps --
connection-name mydcinterconnect --owner-account 123456789012 --interconnect-
id dxcon-fgktov66 --vlan 101

```

Output:

```

{
  "partnerName": "TIVIT",
  "vlan": 101,

```

```
"ownerAccount": "123456789012",
"connectionId": "dxcon-ffzc51m1",
"connectionState": "ordering",
"bandwidth": "500Mbps",
"location": "TIVIT",
"connectionName": "mydcinterconnect",
"region": "sa-east-1"
}
```

- Per API i dettagli, vedere [AllocateConnectionOnInterconnect](#) in AWS CLI Command Reference.

## allocate-hosted-connection

Il seguente esempio di codice mostra come utilizzare `allocate-hosted-connection`.

### AWS CLI

Per creare una connessione ospitata su un'interconnessione

L'esempio seguente crea una connessione ospitata sull'interconnessione specificata.

```
aws directconnect allocate-hosted-connection \
  --bandwidth 500Mbps \
  --connection-name mydcinterconnect \
  --owner-account 123456789012 \
  -connection-id dxcon-fgktov66 \
  -vlan 101
```

Output:

```
{
  "partnerName": "TIVIT",
  "vlan": 101,
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-ffzc51m1",
  "connectionState": "ordering",
  "bandwidth": "500Mbps",
  "location": "TIVIT",
  "connectionName": "mydcinterconnect",
  "region": "sa-east-1"
}
```

```
}

```

- Per API i dettagli, vedere [AllocateHostedConnection](#) in AWS CLI Command Reference.

## allocate-private-virtual-interface

Il seguente esempio di codice mostra come utilizzare `allocate-private-virtual-interface`.

### AWS CLI

Per fornire un'interfaccia virtuale privata

Il `allocate-private-virtual-interface` comando seguente fornisce un'interfaccia virtuale privata di proprietà di un cliente diverso:

```
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
```

Output:

```
{
  "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"
}
```



- Per API i dettagli, vedere [AllocatePrivateVirtualInterface](#) in AWS CLI Command Reference.

## allocate-public-virtual-interface

Il seguente esempio di codice mostra come utilizzare `allocate-public-virtual-interface`.

### AWS CLI

Per fornire un'interfaccia virtuale pubblica

Il `allocate-public-virtual-interface` comando seguente fornisce un'interfaccia virtuale pubblica di proprietà di un altro cliente:

```
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]
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [AllocatePublicVirtualInterface](#) in AWS CLI Command Reference.

## allocate-transit-virtual-interface

Il seguente esempio di codice mostra come utilizzare `allocate-transit-virtual-interface`.

### AWS CLI

Fornire un'interfaccia virtuale di transito di proprietà dell' AWS account specificato

L'`allocate-transit-virtual-interface` esempio seguente fornisce un'interfaccia virtuale di transito per l'account specificato.

```
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=0xzxcgA9YoW9h58u8SEXAMPLE, amazonAddress=192.16"
```

Output:

```
{
  "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": "0xzxcgA9YoW9h58u8SEXAMPLE",
```

```

    "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>0xzxcgA9YoW9h58u8EXAMPLE</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": "0xzxcgA9YoW9h58u8EXAMPLE",
        "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, vedere [Creazione di un'interfaccia virtuale Hosted Transit](#) nella Guida per l'utente di AWS Direct Connect.

- Per API i dettagli, vedere [AllocateTransitVirtualInterface](#) in AWS CLI Command Reference.

## associate-connection-with-lag

Il seguente esempio di codice mostra come utilizzare `associate-connection-with-lag`.

### AWS CLI

Per associare una connessione a LAG

L'esempio seguente associa la connessione specificata a quella specificata LAG.

Comando:

```
aws directconnect associate-connection-with-lag --lag-id dxlag-fhccu14t --  
connection-id dxcon-fg9607vm
```

Output:

```
{  
  "ownerAccount": "123456789012",  
  "connectionId": "dxcon-fg9607vm",  
  "lagId": "dxlag-fhccu14t",  
  "connectionState": "requested",  
  "bandwidth": "1Gbps",  
  "location": "EqDC2",  
  "connectionName": "Con2ForLag",  
  "region": "us-east-1"  
}
```

- Per API i dettagli, vedere [AssociateConnectionWithLag](#) in AWS CLI Command Reference.

## associate-hosted-connection

Il seguente esempio di codice mostra come utilizzare `associate-hosted-connection`.

### AWS CLI

Per associare una connessione ospitata a LAG

L'esempio seguente associa la connessione ospitata specificata a quella specificata LAG.

Comando:

```
aws directconnect associate-hosted-connection --parent-connection-id dxlag-fhccu14t
--connection-id dxcon-fg9607vm
```

Output:

```
{
  "partnerName": "TIVIT",
  "vlan": 101,
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "lagId": "dxlag-fhccu14t",
  "connectionState": "ordering",
  "bandwidth": "500Mbps",
  "location": "TIVIT",
  "connectionName": "mydcinterconnect",
  "region": "sa-east-1"
}
```

- Per API i dettagli, vedere [AssociateHostedConnection](#) in AWS CLI Command Reference.

## associate-virtual-interface

Il seguente esempio di codice mostra come utilizzare `associate-virtual-interface`.

### AWS CLI

Per associare un'interfaccia virtuale a una connessione

L'esempio seguente associa l'interfaccia virtuale specificata a quella specificata LAG. In alternativa, per associare l'interfaccia virtuale a una connessione, specificare l'ID di una connessione AWS Direct Connect per `--connection-id`; ad esempio, `dxcon-ffnikghc`.

Comando:

```
aws directconnect associate-virtual-interface --connection-id dxlag-ffjhj9lx --
virtual-interface-id dxvif-fgputw0j
```

Output:

```
{
  "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"
}

```

- Per API i dettagli, vedere [AssociateVirtualInterface](#) in AWS CLI Command Reference.

## confirm-connection

Il seguente esempio di codice mostra come utilizzare `confirm-connection`.

### AWS CLI

Per confermare la creazione di una connessione ospitata su un'interconnessione

Il `confirm-connection` comando seguente conferma la creazione di una connessione ospitata su un'interconnessione:

```
aws directconnect confirm-connection --connection-id dxcon-fg2wi7hy
```

Output:

```
{
  "connectionState": "pending"
}
```

- Per API i dettagli, vedere [ConfirmConnection](#) in AWS CLI Command Reference.

## confirm-private-virtual-interface

Il seguente esempio di codice mostra come utilizzare `confirm-private-virtual-interface`.

### AWS CLI

Accettare la proprietà di un'interfaccia virtuale privata

Il `confirm-private-virtual-interface` comando seguente accetta la proprietà di un'interfaccia virtuale privata creata da un altro cliente:

```
aws directconnect confirm-private-virtual-interface --virtual-interface-id dxvif-fgy8orxu --virtual-gateway-id vgw-e4a47df9
```

Output:

```
{
  "virtualInterfaceState": "pending"
}
```

```
}
```

- Per API i dettagli, vedere [ConfirmPrivateVirtualInterface](#) in AWS CLI Command Reference.

## **confirm-public-virtual-interface**

Il seguente esempio di codice mostra come utilizzare `confirm-public-virtual-interface`.

### AWS CLI

Accettare la proprietà di un'interfaccia virtuale pubblica

Il `confirm-public-virtual-interface` comando seguente accetta la proprietà di un'interfaccia virtuale pubblica creata da un altro cliente:

```
aws directconnect confirm-public-virtual-interface --virtual-interface-id dxvif-fg9xo9vp
```

Output:

```
{  
  "virtualInterfaceState": "verifying"  
}
```

- Per API i dettagli, vedere [ConfirmPublicVirtualInterface](#) in AWS CLI Command Reference.

## **confirm-transit-virtual-interface**

Il seguente esempio di codice mostra come utilizzare `confirm-transit-virtual-interface`.

### AWS CLI

Accettare la proprietà di un'interfaccia virtuale di transito

Quanto segue `confirm-transit-virtual-interface` accetta la proprietà di un'interfaccia virtuale di transito creata da un altro cliente.

```
aws directconnect confirm-transit-virtual-interface \  
  --virtual-interface-id dxvif-fEXAMPLE \  
  --direct-connect-gateway-id 4112ccf9-25e9-4111-8237-b6c5dEXAMPLE
```



**Output:**

```
{
  "virtualInterfaceState": "pending"
}
```

Per ulteriori informazioni, vedere [Accettazione di un'interfaccia virtuale ospitata](#) nella Guida per l'utente di AWS Direct Connect.

- Per API i dettagli, vedere [ConfirmTransitVirtualInterface](#) in AWS CLI Command Reference.

**create-bgp-peer**

Il seguente esempio di codice mostra come utilizzare `create-bgp-peer`.

**AWS CLI**

Per creare una sessione di IPv6 BGP peering

L'esempio seguente crea una sessione di IPv6 BGP peering su un'interfaccia virtuale privata. `dxvif-fg1vuj3d` Gli IPv6 indirizzi peer vengono assegnati automaticamente da Amazon.

Comando:

```
aws directconnect create-bgp-peer --virtual-interface-id dxvif-fg1vuj3d --new-bgp-peer asn=64600,addressFamily=ipv6
```

**Output:**

```
{
  "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_ukbCer16EYA0example",
  }
}
```

```

"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"
}
}

```

- Per API i dettagli, consulta AWS CLI Command [CreateBgpPeerReference](#).

## create-connection

Il seguente esempio di codice mostra come utilizzare `create-connection`.

## AWS CLI

Per creare una connessione dalla rete a una posizione AWS Direct Connect

Il `create-connection` comando seguente crea una connessione dalla rete a una posizione AWS Direct Connect:

```
aws directconnect create-connection --location TIVIT --bandwidth 1Gbps --connection-name "Connection to AWS"
```

Output:

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg31dyv6",
  "connectionState": "requested",
  "bandwidth": "1Gbps",
  "location": "TIVIT",
  "connectionName": "Connection to AWS",
  "region": "sa-east-1"
}
```

- Per API i dettagli, vedere [CreateConnection](#) in AWS CLI Command Reference.

## **create-direct-connect-gateway-association-proposal**

Il seguente esempio di codice mostra come utilizzare `create-direct-connect-gateway-association-proposal`.

### AWS CLI

Per creare una proposta per associare il gateway di transito specificato al gateway Direct Connect specificato

L'`create-direct-connect-gateway-association-proposal` esempio seguente crea una proposta che associa il gateway di transito specificato al gateway Direct Connect specificato.

```
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
```

Output:

```
{
  "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"
      }
    ]
  }
}
```

Per ulteriori informazioni, vedere [Creazione di una proposta di associazione Transit Gateway](#) nella Guida per l'utente di AWS Direct Connect.

- Per API i dettagli, vedere [CreateDirectConnectGatewayAssociationProposal](#) in AWS CLI Command Reference.

## create-direct-connect-gateway-association

Il seguente esempio di codice mostra come utilizzare `create-direct-connect-gateway-association`.

### AWS CLI

Per associare un gateway privato virtuale a un gateway Direct Connect

L'esempio seguente associa il gateway privato virtuale al gateway `vgw-6efe725e` Direct Connect. `5f294f92-bafb-4011-916d-9b0bexample` È necessario eseguire il comando nella regione in cui si trova il gateway privato virtuale.

Comando:

```
aws directconnect create-direct-connect-gateway-association --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample --virtual-gateway-id vgw-6efe725e
```

Output:

```
{
  "directConnectGatewayAssociation": {
    "associationState": "associating",
    "virtualGatewayOwnerAccount": "123456789012",
    "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
    "virtualGatewayId": "vgw-6efe725e",
    "virtualGatewayRegion": "us-east-2"
  }
}
```

- Per API i dettagli, vedere [CreateDirectConnectGatewayAssociation](#) in AWS CLI Command Reference.

## create-direct-connect-gateway

Il seguente esempio di codice mostra come utilizzare `create-direct-connect-gateway`.

AWS CLI

Per creare un gateway Direct Connect

L'esempio seguente crea un gateway Direct Connect con il nome `DxGateway1`.

Comando:

```
aws directconnect create-direct-connect-gateway --direct-connect-gateway-name "DxGateway1"
```

Output:

```
{
  "directConnectGateway": {
    "amazonSideAsn": 64512,
    "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bdexample",
    "ownerAccount": "123456789012",
  }
}
```

```
"directConnectGatewayName": "DxGateway1",
"directConnectGatewayState": "available"
}
}
```

- Per API i dettagli, vedere [CreateDirectConnectGateway](#) in AWS CLI Command Reference.

## create-interconnect

Il seguente esempio di codice mostra come utilizzare `create-interconnect`.

### AWS CLI

Per creare un'interconnessione tra la rete di un partner e AWS

Il `create-interconnect` comando seguente crea un'interconnessione tra la rete di un partner AWS Direct Connect e una posizione AWS Direct Connect specifica:

```
aws directconnect create-interconnect --interconnect-name "1G Interconnect to AWS"
--bandwidth 1Gbps --location TIVIT
```

Output:

```
{
  "region": "sa-east-1",
  "bandwidth": "1Gbps",
  "location": "TIVIT",
  "interconnectName": "1G Interconnect to AWS",
  "interconnectId": "dxcon-fgktov66",
  "interconnectState": "requested"
}
```

- Per API i dettagli, vedere [CreateInterconnect](#) in AWS CLI Command Reference.

## create-lag

Il seguente esempio di codice mostra come utilizzare `create-lag`.

### AWS CLI

Per creare un LAG con nuove connessioni

L'esempio seguente crea un LAG e richiede due nuove connessioni AWS Direct Connect per la LAG con una larghezza di banda di 1 Gbps.

Comando:

```
aws directconnect create-lag --location CSVA1 --number-of-connections 2 --  
connections-bandwidth 1Gbps --lag-name 1GBLag
```

Output:

```
{  
  "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"  
}
```

```
}
```

Per creare una connessione LAG utilizzando una connessione esistente

L'esempio seguente crea una connessione LAG da una connessione esistente nell'account e richiede una seconda nuova connessione LAG con la stessa larghezza di banda e la stessa posizione della connessione esistente.

Comando:

```
aws directconnect create-lag --location EqDC2 --number-of-connections 2 --  
connections-bandwidth 1Gbps --lag-name 2ConnLAG --connection-id dxcon-fgk145dr
```

Output:

```
{  
  "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"
  }

```

- Per API i dettagli, vedere [CreateLag](#) in AWS CLI Command Reference.

## create-private-virtual-interface

Il seguente esempio di codice mostra come utilizzare `create-private-virtual-interface`.

### AWS CLI

Per creare un'interfaccia virtuale privata

Il `create-private-virtual-interface` comando seguente crea un'interfaccia virtuale privata:

```

aws directconnect create-private-virtual-interface --connection-id dxcon-ffjrkx17 --
new-private-virtual-
interface virtualInterfaceName=PrivateVirtualInterface,vlan=101,asn=65000,authKey=asdf34exam
aba37db6

```

Output:

```

{
  "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"
}
```

- Per API i dettagli, vedere [CreatePrivateVirtualInterface](#) in AWS CLI Command Reference.

## create-public-virtual-interface

Il seguente esempio di codice mostra come utilizzare `create-public-virtual-interface`.

### AWS CLI

Per creare un'interfaccia virtuale pubblica

Il `create-public-virtual-interface` comando seguente crea un'interfaccia virtuale pubblica:

```
aws directconnect create-public-virtual-interface --connection-id dxcon-ffjrkx17 --
new-public-virtual-
interface virtualInterfaceName=PublicVirtualInterface,vlan=2000,asn=65000,authKey=asdf34exam
{cidr=203.0.113.4/30}
```

Output:

```
{
  "virtualInterfaceState": "verifying",
  "asn": 65000,
  "vlan": 2000,
  "customerAddress": "203.0.113.2/30",
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-ffjrkx17",
  "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"
}

```

- Per API i dettagli, vedere [CreatePublicVirtualInterface](#) in AWS CLI Command Reference.

## create-transit-virtual-interface

Il seguente esempio di codice mostra come utilizzare `create-transit-virtual-interface`.

### AWS CLI

Per creare un'interfaccia virtuale di transito

L'`create-transit-virtual-interface` esempio seguente crea un'interfaccia virtuale di transito per la connessione specificata.

```

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}]"

```

Output:

```

{
  "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": "0xzxgA9YoW9h58u8SEXAMPLE",
    "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>0xzxgA9YoW9h58u8Sv0mXRTw</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": "0xzxgA9YoW9h58u8SEXAMPLE",
        "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"
      }
    ]
  }
}

```

```
}
```

Per ulteriori informazioni, vedere [Creazione di un'interfaccia virtuale di transito per il gateway Direct Connect](#) nella Guida per l'utente di AWS Direct Connect.

- Per API i dettagli, vedere [CreateTransitVirtualInterface](#) in AWS CLI Command Reference.

## delete-bgp-peer

Il seguente esempio di codice mostra come utilizzare `delete-bgp-peer`.

### AWS CLI

Per eliminare un BGP peer da un'interfaccia virtuale

L'esempio seguente elimina il IPv6 BGP peer dall'interfaccia virtuale. `dxvif-fg1vuj3d`

Comando:

```
aws directconnect delete-bgp-peer --virtual-interface-id dxvif-fg1vuj3d --asn 64600  
--customer-address 2001:db8:1100:2f0:0:1:9cb4:4216/125
```

Output:

```
{  
  "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_ukbCer16EYA0example",  
    "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"
  }
}

```

- Per API i dettagli, vedere [DeleteBgpPeer](#) in AWS CLI Command Reference.

## delete-connection

Il seguente esempio di codice mostra come utilizzare `delete-connection`.

### AWS CLI

Per eliminare una connessione

Il `delete-connection` comando seguente elimina la connessione specificata:

```
aws directconnect delete-connection --connection-id dxcon-fg31dyv6
```

Output:

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg31dyv6",
  "connectionState": "deleted",
  "bandwidth": "1Gbps",
  "location": "TIVIT",
  "connectionName": "Connection to AWS",
  "region": "sa-east-1"
}
```

- Per API i dettagli, vedere [DeleteConnection](#) in AWS CLI Command Reference.

## delete-direct-connect-gateway-association

Il seguente esempio di codice mostra come utilizzare `delete-direct-connect-gateway-association`.

### AWS CLI

Per eliminare un'associazione gateway Direct Connect

L'`delete-direct-connect-gateway-association` esempio seguente elimina l'associazione del gateway Direct Connect con un gateway di transito con l'ID di associazione specificato.

```
aws directconnect delete-direct-connect-gateway-association --association-id
be85116d-46eb-4b43-a27a-da0c2ad648de
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Associating and Dissociating Transit Gateway](#) nella AWS Direct Connect User Guide.

- Per i API dettagli, vedere [DeleteDirectConnectGatewayAssociation](#) in Command Reference. AWS CLI

## delete-direct-connect-gateway

Il seguente esempio di codice mostra come utilizzare `delete-direct-connect-gateway`.

### AWS CLI

Per eliminare un gateway Direct Connect

L'esempio seguente elimina il gateway `5f294f92-bafb-4011-916d-9b0bexample` Direct Connect.

Comando:

```
aws directconnect delete-direct-connect-gateway --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

Output:

```
{
  "directConnectGateway": {
    "amazonSideAsn": 64512,
    "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
    "ownerAccount": "123456789012",
    "directConnectGatewayName": "DxGateway1",
    "directConnectGatewayState": "deleting"
  }
}
```

- Per API i dettagli, vedere [DeleteDirectConnectGateway](#) in AWS CLI Command Reference.



## delete-interconnect

Il seguente esempio di codice mostra come utilizzare `delete-interconnect`.

### AWS CLI

Per eliminare un'interconnessione

Il `delete-interconnect` comando seguente elimina l'interconnessione specificata:

```
aws directconnect delete-interconnect --interconnect-id dxcon-fgktov66
```

Output:

```
{
  "interconnectState": "deleted"
}
```

- Per API i dettagli, vedere [DeleteInterconnect](#) in AWS CLI Command Reference.

## delete-lag

Il seguente esempio di codice mostra come utilizzare `delete-lag`.

### AWS CLI

Per eliminare un LAG

L'esempio seguente elimina il valore specificato LAG.

Comando:

```
aws directconnect delete-lag --lag-id dxlag-ffrhowd9
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [DeleteLag](#) in AWS CLI Command Reference.

## delete-virtual-interface

Il seguente esempio di codice mostra come utilizzare `delete-virtual-interface`.

### AWS CLI

Per eliminare un'interfaccia virtuale

Il `delete-virtual-interface` comando seguente elimina l'interfaccia virtuale specificata:

```
aws directconnect delete-virtual-interface --virtual-interface-id dxvif-ffhkh74f
```

Output:

```
{
  "virtualInterfaceState": "deleting"
}
```

- Per API i dettagli, vedere [DeleteVirtualInterface](#) in AWS CLI Command Reference.

## describe-connection-loa

Il seguente esempio di codice mostra come utilizzare `describe-connection-loa`.

### AWS CLI

Per descrivere il tuo LOA - CFA per una connessione tramite Linux o Mac OS X

L'esempio seguente descrive la tua connessione LOA - CFA per `dxcon-fh6ayh1d`. Il contenuto di LOA - è codificato in CFA base64. Questo comando utilizza i `--query` parametri `--output` and per controllare l'output ed estrarre il contenuto della struttura. `loaContent` La parte finale del comando decodifica il contenuto utilizzando l'`base64` utilità e invia l'output a un PDF file.

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --  
output text --query Loa.LoaContent|base64 --decode > myLoaCfa.pdf
```

Per descrivere il tuo LOA - CFA per una connessione tramite Windows

L'esempio precedente richiede l'uso dell'`base64` utilità per decodificare l'output. Su un computer Windows, è possibile utilizzare `certutil` invece. Nel seguente esempio, il primo comando descrive la connessione LOA - CFA for `dxcon-fh6ayh1d` e utilizza i `--query` parametri `--output` and per controllare l'output ed estrarre il contenuto della `LoaContent` struttura in un file chiamato `myLoaCfa.base64`. Il secondo comando utilizza l'`certutil` utilità per decodificare il file e inviare l'output a un PDF file.

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --  
output text --query Loa.LoaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

Per ulteriori informazioni sul controllo dell' AWS CLI output, vedere [Controllo dell'output dei comandi dall'interfaccia a riga di AWS comando nella Guida per l'utente dell'interfaccia a riga di AWS comando](#).

- Per API i dettagli, vedere [DescribeConnectionLoa](#) in AWS CLI Command Reference.

## describe-connections-on-interconnect

Il seguente esempio di codice mostra come utilizzare `describe-connections-on-interconnect`.

### AWS CLI

Per elencare le connessioni su un'interconnessione

Il `describe-connections-on-interconnect` comando seguente elenca le connessioni che sono state fornite sull'interconnessione specificata:

```
aws directconnect describe-connections-on-interconnect --interconnect-id dxcon-  
fgktov66
```

Output:

```
{
  "connections": [
    {
      "partnerName": "TIVIT",
      "vlan": 101,
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-ffzc51m1",
      "connectionState": "ordering",
      "bandwidth": "500Mbps",
      "location": "TIVIT",
      "connectionName": "mydcinterconnect",
      "region": "sa-east-1"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeConnectionsOnInterconnect](#) in AWS CLI Command Reference.

## describe-connections

Il seguente esempio di codice mostra come utilizzare `describe-connections`.

### AWS CLI

Per elencare tutte le connessioni nella regione corrente

Il `describe-connections` comando seguente elenca tutte le connessioni nella regione corrente:

```
aws directconnect describe-connections
```

Output:

```
{
  "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"
    }
]
}

```

- Per API i dettagli, vedere [DescribeConnections](#) in AWS CLI Command Reference.

## describe-direct-connect-gateway-association-proposals

Il seguente esempio di codice mostra come utilizzare `describe-direct-connect-gateway-association-proposals`.

### AWS CLI

Per descrivere le vostre proposte di associazione al gateway Direct Connect

L'esempio seguente mostra i dettagli sulle proposte di associazione del gateway Direct Connect.

```
aws directconnect describe-direct-connect-gateway-association-proposals
```

Output:

```

{
  "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"
    }
  ]
}
]
```

Per ulteriori informazioni, vedere [Associating and Dissociating Transit Gateway](#) nella AWS Direct Connect User Guide.

- Per i API dettagli, vedere [DescribeDirectConnectGatewayAssociationProposals](#) in Command Reference.AWS CLI

## describe-direct-connect-gateway-associations

Il seguente esempio di codice mostra come utilizzare `describe-direct-connect-gateway-associations`.

### AWS CLI

Per descrivere le associazioni dei gateway Direct Connect

L'esempio seguente descrive tutte le associazioni con il gateway Direct Connect `5f294f92-bafb-4011-916d-9b0bexample`.

Comando:

```
aws directconnect describe-direct-connect-gateway-associations --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

Output:

```
{
  "nextToken":
  "eyJ2IjoxLCJzIjoxLCJpIjoiOU83OTFodzdycnZCbkn4MExHeHVwQT09IiwiaWYyI6InIxTEN0UEVHV0I1UF1kaWFnN1",
  "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"
    }
  ]
}
```

```
}
```

- Per API i dettagli, vedere [DescribeDirectConnectGatewayAssociations](#) in AWS CLI Command Reference.

## describe-direct-connect-gateway-attachments

Il seguente esempio di codice mostra come utilizzare `describe-direct-connect-gateway-attachments`.

### AWS CLI

Per descrivere gli allegati del gateway Direct Connect

L'esempio seguente descrive le interfacce virtuali collegate al gateway `5f294f92-bafb-4011-916d-9b0bexample` Direct Connect.

Comando:

```
aws directconnect describe-direct-connect-gateway-attachments --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

Output:

```
{
  "directConnectGatewayAttachments": [
    {
      "virtualInterfaceOwnerAccount": "123456789012",
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
      "virtualInterfaceRegion": "us-east-2",
      "attachmentState": "attaching",
      "virtualInterfaceId": "dxvif-fg9zyabc"
    }
  ],
  "nextToken":
  "eyJ2IjoxLCJzIjoxLCJpIjoibEhXd1NpUXF5RzhoL1JyUW52S1V2QT09IiwieyYyI6Im5wQjFHQ0RyQUdRS3puNnNXcU"
}
```

- Per API i dettagli, vedere [DescribeDirectConnectGatewayAttachments](#) in AWS CLI Command Reference.



## describe-direct-connect-gateways

Il seguente esempio di codice mostra come utilizzare `describe-direct-connect-gateways`.

### AWS CLI

Per descrivere i gateway Direct Connect

L'esempio seguente descrive tutti i gateway Direct Connect.

Comando:

```
aws directconnect describe-direct-connect-gateways
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeDirectConnectGateways](#) in AWS CLI Command Reference.

## describe-hosted-connections

Il seguente esempio di codice mostra come utilizzare `describe-hosted-connections`.

## AWS CLI

Per elencare le connessioni su un'interconnessione

L'esempio seguente elenca le connessioni che sono state fornite su una determinata interconnessione.

Comando:

```
aws directconnect describe-hosted-connections --connection-id dxcon-fgktov66
```

Output:

```
{
  "connections": [
    {
      "partnerName": "TIVIT",
      "vlan": 101,
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-ffzc51m1",
      "connectionState": "ordering",
      "bandwidth": "500Mbps",
      "location": "TIVIT",
      "connectionName": "mydcinterconnect",
      "region": "sa-east-1"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeHostedConnections](#) in AWS CLI Command Reference.

## describe-interconnect-loa

Il seguente esempio di codice mostra come utilizzare `describe-interconnect-loa`.

## AWS CLI

Per descrivere il vostro LOA - CFA per un'interconnessione tramite Linux o Mac OS X

L'esempio seguente descrive il tuo LOA - CFA for `dxcon-fh6ayh1d` interconnect. Il contenuto di LOA - è codificato in CFA base64. Questo comando utilizza i `--query` parametri `--output`

and per controllare l'output ed estrarre il contenuto della struttura. `loaContent` La parte finale del comando decodifica il contenuto utilizzando l'`base64` utilità e invia l'output a un PDF file.

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --  
output text --query loa.loaContent|base64 --decode > myLoaCfa.pdf
```

Per descrivere il tuo LOA - CFA per un'interconnessione tramite Windows

L'esempio precedente richiede l'uso dell'`base64` utilità per decodificare l'output. Su un computer Windows, è possibile utilizzare `certutil` invece. Nel seguente esempio, il primo comando descrive il comando LOA - CFA for interconnect `dxcon-fh6ayh1d` e utilizza i `--query` parametri `--output` and per controllare l'output ed estrarre il contenuto della `loaContent` struttura in un file chiamato `myLoaCfa.base64`. Il secondo comando utilizza l'`certutil` utilità per decodificare il file e inviare l'output a un PDF file.

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --  
output text --query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

Per ulteriori informazioni sul controllo dell' AWS CLI output, vedere [Controllo dell'output dei comandi dall'interfaccia a riga di AWS comando nella Guida per l'utente dell'interfaccia a riga di AWS comando](#).

- Per API i dettagli, vedere [DescribeInterconnectLoa](#) in AWS CLI Command Reference.

## describe-interconnects

Il seguente esempio di codice mostra come utilizzare `describe-interconnects`.

### AWS CLI

Per elencare le interconnessioni

Il `describe-interconnects` comando seguente elenca le interconnessioni di proprietà dell'account AWS :

```
aws directconnect describe-interconnects
```

**Output:**

```
{
  "interconnects": [
    {
      "region": "sa-east-1",
      "bandwidth": "1Gbps",
      "location": "TIVIT",
      "interconnectName": "1G Interconnect to AWS",
      "interconnectId": "dxcon-fgktov66",
      "interconnectState": "down"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeInterconnects](#) in AWS CLI Command Reference.

**describe-lags**

Il seguente esempio di codice mostra come utilizzare `describe-lags`.

**AWS CLI**

Per descrivere il tuo LAGs

Il comando seguente descrive tutti i dati LAGs relativi alla regione corrente.

Comando:

```
aws directconnect describe-lags
```

**Output:**

```
{
  "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"
  }
]
}

```

- Per API i dettagli, vedere [DescribeLags](#) in AWS CLI Command Reference.

## describe-loa

Il seguente esempio di codice mostra come utilizzare `describe-loa`.

### AWS CLI

Per descrivere il tuo LOA - CFA per una connessione tramite Linux o Mac OS X

L'esempio seguente descrive la tua connessione LOA - CFA per `dxcon-fh6ayh1d`. Il contenuto di LOA - è codificato in CFA base64. Questo comando utilizza i `--query` parametri `--output`

and per controllare l'output ed estrarre il contenuto della struttura. `LoaContent` La parte finale del comando decodifica il contenuto utilizzando l'base64utilità e invia l'output a un PDF file.

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --  
query Loa.LoaContent/base64 --decode > myLoaCfa.pdf
```

Per descrivere il tuo LOA - CFA per una connessione tramite Windows

L'esempio precedente richiede l'uso dell'base64utilità per decodificare l'output. Su un computer Windows, è possibile utilizzare `certutil` invece. Nel seguente esempio, il primo comando descrive la connessione LOA - CFA for `dxcon-fh6ayh1d` e utilizza i `--query` parametri `--output` and per controllare l'output ed estrarre il contenuto della `LoaContent` struttura in un file chiamato `myLoaCfa.base64`. Il secondo comando utilizza l'`certutil` utilità per decodificare il file e inviare l'output a un PDF file.

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --  
query Loa.LoaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

Per ulteriori informazioni sul controllo dell' AWS CLI output, vedere [Controllo dell'output dei comandi dall'interfaccia a riga di AWS comando nella Guida per l'utente dell'interfaccia a riga di AWS comando](#).

- Per API i dettagli, vedere [DescribeLoa](#) in AWS CLI Command Reference.

## describe-locations

Il seguente esempio di codice mostra come utilizzare `describe-locations`.

### AWS CLI

Per elencare i partner e le sedi di AWS Direct Connect

Il `describe-locations` comando seguente elenca i partner e le sedi di AWS Direct Connect nella regione corrente:

```
aws directconnect describe-locations
```

Output:

```
{
  "locations": [
    {
      "locationName": "NAP do Brasil, Barueri, Sao Paulo",
      "locationCode": "TNDB"
    },
    {
      "locationName": "Tivit - Site Transamerica (Sao Paulo)",
      "locationCode": "TIVIT"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeLocations](#) in AWS CLI Command Reference.

## describe-tags

Il seguente esempio di codice mostra come utilizzare `describe-tags`.

### AWS CLI

Per descrivere i tag per le tue risorse AWS Direct Connect

Il comando seguente descrive i tag per la connessione `dxcon-abcabc12`.

Comando:

```
aws directconnect describe-tags --resource-arns arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12
```

Output:

```
{
  "resourceTags": [
    {
      "resourceArn": "arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12",
      "tags": [
        {
          "value": "VAConnection",
          "key": "Name"
        }
      ]
    }
  ]
}
```

```
    ]
  }
]
}
```

- Per API i dettagli, vedere [DescribeTags](#) in AWS CLI Command Reference.

## describe-virtual-gateways

Il seguente esempio di codice mostra come utilizzare `describe-virtual-gateways`.

### AWS CLI

Per elencare i gateway privati virtuali

Il `describe-virtual-gateways` comando seguente elenca i gateway privati virtuali di proprietà dell'account AWS :

```
aws directconnect describe-virtual-gateways
```

Output:

```
{
  "virtualGateways": [
    {
      "virtualGatewayId": "vgw-aba37db6",
      "virtualGatewayState": "available"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeVirtualGateways](#) in AWS CLI Command Reference.

## describe-virtual-interfaces

Il seguente esempio di codice mostra come utilizzare `describe-virtual-interfaces`.

### AWS CLI

Per elencare tutte le interfacce virtuali



Il `describe-virtual-interfaces` comando seguente elenca le informazioni su tutte le interfacce virtuali associate all'account AWS :

```
aws directconnect describe-virtual-interfaces --connection-id dxcon-ffjrkx17
```

Output:

```
{
  "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\\\"?>\\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"
    },
    {
      "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"
  }
]
}

```

- Per API i dettagli, vedere [DescribeVirtualInterfaces](#) in AWS CLI Command Reference.

## disassociate-connection-from-lag

Il seguente esempio di codice mostra come utilizzare `disassociate-connection-from-lag`.

### AWS CLI

Per dissociare una connessione da un LAG

L'esempio seguente dissocia la connessione specificata da quella specificata. LAG

Comando:

```
aws directconnect disassociate-connection-from-lag --lag-id dxlag-fhccu14t --
connection-id dxcon-fg9607vm
```

Output:

```
{
  "ownerAccount": "123456789012",
```

```
"connectionId": "dxcon-fg9607vm",
"connectionState": "requested",
"bandwidth": "1Gbps",
"location": "EqDC2",
"connectionName": "Con2ForLag",
"region": "us-east-1"
}
```

- Per API i dettagli, vedere [DisassociateConnectionFromLag](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere un tag a una risorsa AWS Direct Connect

Il comando seguente aggiunge un tag con una chiave `Name` e un valore di `VAConnection` alla connessione `dxcon-abcabc12`. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws directconnect tag-resource --resource-arn arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12 --tags "key=Name,value=VAConnection"
```

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da una risorsa AWS Direct Connect

Il comando seguente rimuove il tag con la chiave `Name` dalla connessione `dxcon-abcabc12`. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws directconnect untag-resource --resource-arn arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-abcabc12 --tag-keys Name
```

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-direct-connect-gateway-association

Il seguente esempio di codice mostra come utilizzare `update-direct-connect-gateway-association`.

### AWS CLI

Per aggiornare gli attributi specificati dell'associazione gateway Direct Connect

L'`update-direct-connect-gateway-association` esempio seguente aggiunge il CIDR blocco specificato a un'associazione gateway Direct Connect.

```
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
```

Output:

```
{
  "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"
      }
    ]
  }
}
```

```
    ]
  }
}
```

Per ulteriori informazioni, consulta [Lavorare con i gateway Direct Connect](#) nella Guida per l'utente di AWS Direct Connect.

- Per API i dettagli, vedere [UpdateDirectConnectGatewayAssociation](#) in AWS CLI Command Reference.

## update-lag

Il seguente esempio di codice mostra come utilizzare `update-lag`.

### AWS CLI

Per aggiornare un LAG

L'esempio seguente modifica il nome del file specificato LAG.

Comando:

```
aws directconnect update-lag --lag-id dxlag-ffjhj9lx --lag-name 2ConnLag
```

Output:

```
{
  "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"
}

```

- Per API i dettagli, vedere [UpdateLag](#) in AWS CLI Command Reference.

## update-virtual-interface-attributes

Il seguente esempio di codice mostra come utilizzare `update-virtual-interface-attributes`.

### AWS CLI

Per aggiornare MTU l'interfaccia virtuale

L'`update-virtual-interface-attributes` esempio seguente aggiorna MTU l'interfaccia virtuale specificata.

```

aws directconnect update-virtual-interface-attributes \
  --virtual-interface-id dxvif-fEXAMPLE \
  --mtu 1500

```

Output:

```

{
  "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": []
}

```

Per ulteriori informazioni, vedere [Impostazione della rete MTU per le interfacce virtuali private o le interfacce virtuali di transito nella Guida](#) per l'utente di AWS Direct Connect.

- Per API i dettagli, vedere [UpdateVirtualInterfaceAttributes](#) in AWS CLI Command Reference.

## AWS Directory Service esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Directory Service.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **describe-directories**

Il seguente esempio di codice mostra come utilizzare `describe-directories`.

AWS CLI

Per ottenere dettagli sulle tue directory

L'esempio seguente mostra i dettagli sulla directory specificata.

```
aws ds describe-directories \  
  --directory-id d-a1b2c3d4e5
```

Output:

```
{  
  "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",
      "DnsIpAddrs": [
        "203.113.0.248",
        "203.113.0.253"
      ],
      "VpcSettings": {
        "VpcId": "vpc-a1b2c3d4",
        "SubnetIds": [
          "subnet-a1b2c3d4",
          "subnet-d4c3b2a1"
        ],
        "AvailabilityZones": [
          "us-west-2a",
          "us-west-2c"
        ]
      }
    }
  }
}
]
}

```

- Per API i dettagli, vedere [DescribeDirectories](#) in AWS CLI Command Reference.

## describe-trusts

Il seguente esempio di codice mostra come utilizzare `describe-trusts`.

### AWS CLI

Per ottenere dettagli sui vostri rapporti di fiducia

L'output di `aws ds describe-trusts` seguente visualizza i dettagli sulle relazioni di trust per la directory specificata.

```
aws ds describe-trusts \
  --directory-id d-a1b2c3d4e5
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeTrusts](#) in AWS CLI Command Reference.

## AWS DMS esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS DMS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti



```
--s3-settings file://s3-settings.json
```

Contenuto di `s3-settings.json`.

```
{
  "BucketName": "my-corp-data",
  "BucketFolder": "sourcedata",
  "ServiceAccessRoleArn": "arn:aws:iam::123456789012:role/my-s3-access-role"
}
```

Output:

```
{
  "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
    }
  }
}
```

Per ulteriori informazioni, vedere [Working with AWS DMS Endpoints](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [CreateEndpoint](#) in AWS CLI Command Reference.

## create-event-subscription

Il seguente esempio di codice mostra come utilizzare `create-event-subscription`.

### AWS CLI

Per elencare le sottoscrizioni agli eventi

L'esempio seguente crea una sottoscrizione a un evento a un SNS argomento Amazon (`my-sns-topic`).

```
aws dms create-event-subscription \
  --subscription-name my-dms-events \
  --sns-topic-arn arn:aws:sns:us-east-1:123456789012:my-sns-topic
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, vedere [Working with Events and Notifications](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [CreateEventSubscription](#) in AWS CLI Command Reference.

## create-replication-instance

Il seguente esempio di codice mostra come utilizzare `create-replication-instance`.

### AWS CLI

Per creare un'istanza di replica

L'esempio seguente crea un'istanza di replica.

```
aws dms create-replication-instance \  
  --replication-instance-identifier my-repl-instance \  
  --replication-instance-class dms.t2.micro \  
  --allocated-storage 5
```

Output:

```
{  
  "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
}
}

```

Per ulteriori informazioni, vedere [Working with an AWS DMS Replication Instance](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [CreateReplicationInstance](#) in AWS CLI Command Reference.

## create-replication-subnet-group

Il seguente esempio di codice mostra come utilizzare `create-replication-subnet-group`.

### AWS CLI

Per creare un gruppo di sottoreti

L'esempio seguente crea un gruppo composto da 3 sottoreti.

```
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
```

Output:

```
{  
  "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"
  }
]
}
}

```

Per ulteriori informazioni, vedere [Configurazione di una rete per un'istanza di replica](#) nella Guida per l'utente del AWS Database Migration Service.

- Per API i dettagli, vedere [CreateReplicationSubnetGroup](#) in AWS CLI Command Reference.

## create-replication-task

Il seguente esempio di codice mostra come utilizzare `create-replication-task`.

### AWS CLI

Per creare un'attività di replica

L'`create-replication-task` esempio seguente crea un'attività di replica.

```

aws dms create-replication-task \
  --replication-task-identifier movedata \
  --source-endpoint-arn arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPMWGAYUVLKIB732KEVWA \
  --target-endpoint-arn arn:aws:dms:us-east-1:123456789012:endpoint:E0M45FKCZEYHZBFGAGZT3QEC5U \
  --replication-instance-arn $RI_ARN \
  --migration-type full-load \
  --table-mappings file://table-mappings.json

```

Contenuto di `table-mappings.json`.

```

{
  "rules": [
    {
      "rule-type": "selection",
      "rule-id": "1",
      "rule-name": "1",
      "object-locator": {
        "schema-name": "prodrep",

```

```

        "table-name": "%"
      },
      "rule-action": "include",
      "filters": []
    }
  ]
}

```

Output:

```

{
  "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": "creating",
    "ReplicationTaskCreationDate": 1590524772.505,
    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
  }
}

```

Per ulteriori informazioni, vedere [Working with AWS DMS Tasks](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [CreateReplicationTask](#) in AWS CLI Command Reference.

## delete-connection

Il seguente esempio di codice mostra come utilizzare `delete-connection`.

### AWS CLI

Per eliminare una connessione

L'`delete-connection` esempio seguente dissocia un endpoint da un'istanza di replica.

```
aws dms delete-connection \  
  --endpoint-arn arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWGWAYUVLKIB732KEVWA \  
  --replication-instance-arn arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE
```

Output:

```
{  
  "Connection": {  
    "ReplicationInstanceArn": "arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",  
    "EndpointArn": "arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWGWAYUVLKIB732KEVWA",  
    "Status": "deleting",  
    "EndpointIdentifier": "src-database-1",  
    "ReplicationInstanceIdentifier": "my-repl-instance"  
  }  
}
```

Per ulteriori informazioni, vedere [https://docs.aws.amazon.com/dms/latest/userguide/CHAP\\_endpoints.creating.html](https://docs.aws.amazon.com/dms/latest/userguide/CHAP_endpoints.creating.html) nella Database Migration AWS Service User Guide.

- Per API i dettagli, vedere in Command Reference. [DeleteConnection](#) AWS CLI

## delete-endpoint

Il seguente esempio di codice mostra come utilizzare `delete-endpoint`.

AWS CLI

Per eliminare un endpoint

L'`delete-endpoint` seguente esempio seguente elimina un endpoint.

```
aws dms delete-endpoint \  
  --endpoint-arn arn:aws:dms:us-east-1:123456789012:endpoint:0UJJVX04XZ4CYTSEG5XGMN2R3Y
```

Output:

```
{
```

```

    "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:0UJJVX04XZ4CYTSEG5XGMN2R3Y",
      "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
      }
    }
  }
}

```

Per ulteriori informazioni, vedere [Working with AWS DMS Endpoints](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DeleteEndpoint](#) in AWS CLI Command Reference.

## delete-event-subscription

Il seguente esempio di codice mostra come utilizzare `delete-event-subscription`.

### AWS CLI

Per eliminare una sottoscrizione a un evento

L'esempio seguente elimina un abbonamento a un SNS argomento Amazon.

```

aws dms delete-event-subscription \
  --subscription-name "my-dms-events"

```

**Output:**

```
{
  "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
  }
}
```

Per ulteriori informazioni, vedere [Working with Events and Notifications](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DeleteEventSubscription](#) in AWS CLI Command Reference.

**delete-replication-instance**

Il seguente esempio di codice mostra come utilizzare `delete-replication-instance`.

**AWS CLI**

Per eliminare un'istanza di replica

Nell'esempio seguente di `delete-replication-instance` viene eliminata un'istanza di replica.

```
aws dms delete-replication-instance \
  --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

**Output:**

```
{
  "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"
}
}

```

Per ulteriori informazioni, vedere [Working with an AWS DMS Replication Instance](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DeleteReplicationInstance](#) in AWS CLI Command Reference.

## delete-replication-subnet-group

Il seguente esempio di codice mostra come utilizzare `delete-replication-subnet-group`.

### AWS CLI

Per eliminare un gruppo di sottoreti

L'`delete-replication-subnet-group`esempio seguente elimina un gruppo di sottoreti.

```
aws dms delete-replication-subnet-group \  
--replication-subnet-group-identifier my-subnet-group
```

Output:

```
(none)
```

Per ulteriori informazioni, vedere [Configurazione di una rete per un'istanza di replica](#) nella Guida per l'utente del AWS Database Migration Service.

- Per API i dettagli, vedere [DeleteReplicationSubnetGroup](#) in AWS CLI Command Reference.

## delete-replication-task

Il seguente esempio di codice mostra come utilizzare `delete-replication-task`.

### AWS CLI

Per eliminare un'attività di replica

L'`delete-replication-task`esempio seguente elimina un'attività di replica.

```
aws dms delete-replication-task \  
--replication-task-arn arn:aws:dms:us-  
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

Output:

```
{  
  "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": "deleting",
    "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
    "ReplicationTaskCreationDate": 1590524772.505,
    "ReplicationTaskStartDate": 1590789988.677,
    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
  }
}

```

Per ulteriori informazioni, vedere [Working with AWS DMS Tasks](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DeleteReplicationTask](#) in AWS CLI Command Reference.

## describe-account-attributes

Il seguente esempio di codice mostra come utilizzare `describe-account-attributes`.

### AWS CLI

Per descrivere gli attributi dell'account

L'esempio seguente elenca gli attributi del tuo AWS account.

```
aws dms describe-account-attributes
```

Output:

```

{
  "AccountQuotas": [
    {
      "AccountQuotaName": "ReplicationInstances",
      "Used": 1,
      "Max": 20
    },
  ],
}

```

```

    {
      "AccountQuotaName": "AllocatedStorage",
      "Used": 5,
      "Max": 10000
    },
    ...remaining output omitted...

  ],
  "UniqueAccountIdentifier": "cqahfbfy5xee"
}

```

- Per API i dettagli, vedere [DescribeAccountAttributes](#) in AWS CLI Command Reference.

## describe-certificates

Il seguente esempio di codice mostra come utilizzare `describe-certificates`.

### AWS CLI

Per elencare i certificati disponibili

L'esempio seguente elenca i certificati disponibili nel tuo AWS account.

```
aws dms describe-certificates
```

Output:

```

{
  "Certificates": [
    {
      "CertificateIdentifier": "my-cert",
      "CertificateCreationDate": 1543259542.506,
      "CertificatePem": "-----BEGIN CERTIFICATE-----
\nMIID9DCCAtygAwIBAgIBQjANBgkqhkiG9w0BAQ ...U"
      ... remaining output omitted ...
    }
  ]
}

```

Per ulteriori informazioni, vedere [Using SSL](#) in the AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DescribeCertificates](#) in AWS CLI Command Reference.

## describe-connections

Il seguente esempio di codice mostra come utilizzare `describe-connections`.

### AWS CLI

Per descrivere le connessioni

L'`describe-connections` esempio seguente elenca le connessioni testate tra un'istanza di replica e un endpoint.

```
aws dms describe-connections
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Creating Source and Target Endpoint](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DescribeConnections](#) in AWS CLI Command Reference.

## describe-endpoint-types

Il seguente esempio di codice mostra come utilizzare `describe-endpoint-types`.

## AWS CLI

Per elencare i tipi di endpoint disponibili

L'output del comando `aws dms describe-endpoint-types` seguente elenca i tipi di SQL endpoint My disponibili.

```
aws dms describe-endpoint-types \
  --filters "Name=engine-name,Values=mysql"
```

Output:

```
{
  "SupportedEndpointTypes": [
    {
      "EngineName": "mysql",
      "SupportsCDC": true,
      "EndpointType": "source",
      "EngineDisplayName": "MySQL"
    },
    {
      "EngineName": "mysql",
      "SupportsCDC": true,
      "EndpointType": "target",
      "EngineDisplayName": "MySQL"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Working with AWS DMS Endpoints](https://docs.aws.amazon.com/dms/latest/userguide/CHAP_endpoints.html) < [https://docs.aws.amazon.com/dms/latest/userguide/CHAP\\_endpoints.html](https://docs.aws.amazon.com/dms/latest/userguide/CHAP_endpoints.html) > nella Database Migration Service User Guide.AWS

- Per i API dettagli, [DescribeEndpointTypes](#) vedere in Command AWS CLI Reference.

## describe-endpoints

Il seguente esempio di codice mostra come utilizzare `describe-endpoints`.

## AWS CLI

Per descrivere gli endpoint

L'`describe-endpoint` seguente elenca gli endpoint del tuo AWS account.

```
aws dms describe-endpoints
```

Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, vedere [Working with AWS DMS Endpoints](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DescribeEndpoints](#) in AWS CLI Command Reference.

## describe-event-categories

Il seguente esempio di codice mostra come utilizzare `describe-event-categories`.

### AWS CLI

Per descrivere le categorie di eventi

L'`describe-event-categories` esempio seguente elenca le categorie di eventi disponibili.

```
aws dms describe-event-categories
```

Output:

```
{
  "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"
      ]
    }
  ]
}
```

Per ulteriori informazioni, vedere [Working with Events and Notifications](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DescribeEventCategories](#) in AWS CLI Command Reference.

## describe-event-subscriptions

Il seguente esempio di codice mostra come utilizzare `describe-event-subscriptions`.

### AWS CLI

Per descrivere le sottoscrizioni agli eventi

L'esempio seguente elenca le sottoscrizioni agli eventi a un argomento di Amazon SNS.

```
aws dms describe-event-subscriptions
```

Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, vedere [Working with Events and Notifications](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DescribeEventSubscriptions](#) in AWS CLI Command Reference.

## describe-events

Il seguente esempio di codice mostra come utilizzare `describe-events`.

### AWS CLI

Per elencare DMS gli eventi

L'`describe-events` seguente elenca gli eventi che hanno avuto origine da un'istanza di replica.

```
aws dms describe-events \
  --source-type "replication-instance"
```

Output:

```
{
  "Events": [
    {
      "SourceIdentifier": "my-repl-instance",
      "SourceType": "replication-instance",
      "Message": "Replication application shutdown",
      "EventCategories": [],
      "Date": 1590771645.776
    }
  ]
}
```

Per ulteriori informazioni, vedere [Working with Events and Notifications](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DescribeEvents](#) in AWS CLI Command Reference.

## **describe-orderable-replication-instances**

Il seguente esempio di codice mostra come utilizzare `describe-orderable-replication-instances`.

AWS CLI

Per descrivere le istanze di replica ordinabili

L'`describe-orderable-replication-instances` seguente elenca i tipi di istanze di replica che è possibile ordinare.

```
aws dms describe-orderable-replication-instances
```

Output:



```
{
  "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...
  ]
}
```

Per ulteriori informazioni, vedere [Working with an AWS DMS Replication Instance](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DescribeOrderableReplicationInstances](#) in AWS CLI Command Reference.

## **describe-refresh-schemas-status**

Il seguente esempio di codice mostra come utilizzare `describe-refresh-schemas-status`.

### AWS CLI

Per elencare lo stato di aggiornamento di un endpoint

L'`describe-refresh-schemas-status` esempio seguente restituisce lo stato di una precedente richiesta di aggiornamento.

```
aws dms describe-refresh-schemas-status \  
  --endpoint-arn arn:aws:dms:us-  
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA
```

Output:

```
{  
  "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  
  }  
}
```

- Per API i dettagli, vedere [DescribeRefreshSchemasStatus](#) in AWS CLI Command Reference.

## **describe-replication-instances**

Il seguente esempio di codice mostra come utilizzare `describe-replication-instances`.

### AWS CLI

Per descrivere le istanze di replica

L'output di `aws dms describe-replication-instances` seguente elenca le istanze di replica presenti nell'account. AWS

```
aws dms describe-replication-instances
```

Output:

```
{
  "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
  }
]
}

```

Per ulteriori informazioni, vedere [Working with an AWS DMS Replication Instance](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DescribeReplicationInstances](#) in AWS CLI Command Reference.

## describe-replication-subnet-groups

Il seguente esempio di codice mostra come utilizzare `describe-replication-subnet-groups`.

### AWS CLI

Per visualizzare i gruppi di sottoreti disponibili

L'esempio seguente elenca i gruppi di sottoreti disponibili.

```

aws dms describe-replication-subnet-groups \
  --filter "Name=replication-subnet-group-id,Values=my-subnet-group"

```

Output:

```

{
  "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"
  }
]
}
```

Per ulteriori informazioni, vedere [Configurazione di una rete per un'istanza di replica](#) nella Guida per l'utente del AWS Database Migration Service.

- Per API i dettagli, vedere [DescribeReplicationSubnetGroups](#) in AWS CLI Command Reference.

## **describe-replication-task-assessment-results**

Il seguente esempio di codice mostra come utilizzare `describe-replication-task-assessment-results`.

### AWS CLI

Per elencare i risultati delle valutazioni delle attività di replica

L'`describe-replication-task-assessment-results` esempio seguente elenca i risultati di una precedente valutazione delle attività.

```
aws dms describe-replication-task-assessment-results
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Creazione di un rapporto di valutazione delle attività](#) nella Guida per l'utente del AWS Database Migration Service.

- Per API i dettagli, vedere [DescribeReplicationTaskAssessmentResults](#) in AWS CLI Command Reference.

## describe-replication-tasks

Il seguente esempio di codice mostra come utilizzare `describe-replication-tasks`.

### AWS CLI

Per descrivere un'attività di replica

L'`describe-replication-tasks` esempio seguente descrive le attività di replica correnti.

```
aws dms describe-replication-tasks
```

Output:

```
{
  "ReplicationTasks": [
    {
      "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": "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
    }
  }
]
}

```

Per ulteriori informazioni, vedere [Working with AWS DMS Tasks](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DescribeReplicationTasks](#) in AWS CLI Command Reference.

## describe-schemas

Il seguente esempio di codice mostra come utilizzare `describe-schemas`.

### AWS CLI

Per descrivere gli schemi dei database

L'`describe-schemas` esempio seguente elenca le tabelle disponibili in un endpoint.

```
aws dms describe-schemas \
```



```
--endpoint-arn "arn:aws:dms:us-east-1:123456789012:endpoint:6GGI6YPWYGAYUVLKIB732KEVWA"
```

Output:

```
{
  "Schemas": [
    "prodrep"
  ]
}
```

Per ulteriori informazioni, vedere [Questo è il titolo dell'argomento](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [DescribeSchemas](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

AWS CLI

Per elencare i tag di una risorsa

L'`list-tags-for-resource` esempio seguente elenca i tag per un'istanza di replica.

```
aws dms list-tags-for-resource \
  --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE
```

Output:

```
{
  "TagList": [
    {
      "Key": "Project",
      "Value": "dbMigration"
    },
    {
      "Key": "Environment",
      "Value": "PROD"
    }
  ]
}
```

```
}

```

Per ulteriori informazioni, vedere [Tagging Resources](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## modify-endpoint

Il seguente esempio di codice mostra come utilizzare `modify-endpoint`.

### AWS CLI

Per modificare un endpoint

L'esempio seguente aggiunge un attributo di connessione aggiuntivo a un endpoint.

```
aws dms modify-endpoint \
  --endpoint-arn "arn:aws:dms:us-
east-1:123456789012:endpoint:GUVAFG34EECU0J6QVZ56DAHT3U" \
  --extra-connection-attributes "compressionType=GZIP"
```

Output:

```
{
  "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
    }
}
}

```

Per ulteriori informazioni, vedere [Working with AWS DMS Endpoints < https://docs.aws.amazon.com/dms/latest/userguide/CHAP\\_endpoints.html >](https://docs.aws.amazon.com/dms/latest/userguide/CHAP_endpoints.html) nella Database Migration Service User Guide.AWS

- Per i API dettagli, [ModifyEndpoint](#)vedere in Command AWS CLI Reference.

## modify-event-subscription

Il seguente esempio di codice mostra come utilizzare `modify-event-subscription`.

### AWS CLI

Per modificare la sottoscrizione di un evento

L'`modify-event-subscription`esempio seguente modifica il tipo di origine di una sottoscrizione a un evento.

```

aws dms modify-event-subscription \
  --subscription-name "my-dms-events" \
  --source-type replication-task

```

Output:

```

{
  "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
  }
}

```

Per ulteriori informazioni, vedere [Working with Events and Notifications](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [ModifyEventSubscription](#) in AWS CLI Command Reference.

## modify-replication-instance

Il seguente esempio di codice mostra come utilizzare `modify-replication-instance`.

### AWS CLI

Per modificare un'istanza di replica

L'`modify-replication-instance` seguente modifica un'istanza di replica in modo che utilizzi una distribuzione Multi-AZ.

```
aws dms modify-replication-instance \
  --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \
  --multi-az
```

Output:

```
{
  "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...
```

```
}  
}
```

Per ulteriori informazioni, vedere [Working with an AWS DMS Replication Instance](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [ModifyReplicationInstance](#) in AWS CLI Command Reference.

## modify-replication-subnet-group

Il seguente esempio di codice mostra come utilizzare `modify-replication-subnet-group`.

### AWS CLI

Per modificare un gruppo di sottoreti

L'`modify-replication-subnet-group` esempio seguente modifica gli elenchi delle sottoreti associate a un gruppo di sottoreti.

```
aws dms modify-replication-subnet-group \  
  --replication-subnet-group-identifier my-subnet-group \  
  --subnet-id subnet-da327bf6 subnet-bac383e0
```

### Output:

```
{  
  "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"
    }
]
}
}

```

Per ulteriori informazioni, vedere [Configurazione di una rete per un'istanza di replica](#) nella Guida per l'utente del AWS Database Migration Service.

- Per API i dettagli, vedere [ModifyReplicationSubnetGroup](#) in AWS CLI Command Reference.

## modify-replication-task

Il seguente esempio di codice mostra come utilizzare `modify-replication-task`.

### AWS CLI

Per modificare un'attività di replica

L'`modify-replication-task` esempio seguente modifica i mapping delle tabelle per un task.

```

aws dms modify-replication-task \
  --replication-task-arn "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII" \
  --table-mappings file://table-mappings.json

```

Contenuto di `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": []
    }
  ]
}

```

```

    }
  ]
}

```

### Output:

```

{
  "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:T3OM7OUB5NM2LCVZF7JPGJRNUE",
    "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"
  }
}

```

Per ulteriori informazioni, vedere [Working with AWS DMS Tasks](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [ModifyReplicationTask](#) in AWS CLI Command Reference.

## reboot-replication-instance

Il seguente esempio di codice mostra come utilizzare `reboot-replication-instance`.

### AWS CLI

Per riavviare un'istanza di replica

Nell'esempio seguente di `reboot-replication-instance` viene riavviata un'istanza di replica.

```
aws dms reboot-replication-instance \  
  --replication-instance-arn arn:aws:dms:us-  
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE
```

Output:

```
{  
  "ReplicationInstance": {  
    "ReplicationInstanceIdentifier": "my-repl-instance",  
    "ReplicationInstanceClass": "dms.t2.micro",  
    "ReplicationInstanceStatus": "rebooting",  
    "AllocatedStorage": 5,  
    "InstanceCreateTime": 1590011235.952,  
    ... output omitted ...  
  }  
}
```

Per ulteriori informazioni, vedere [Working with an AWS DMS Replication Instance](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [RebootReplicationInstance](#) in AWS CLI Command Reference.

## refresh-schemas

Il seguente esempio di codice mostra come utilizzare `refresh-schemas`.

AWS CLI

Per aggiornare gli schemi del database

L'esempio seguente richiede l'aggiornamento dell'elenco degli schemi in un endpoint.

```
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:6GGI6YPWGWAYUVLKIB732KEVWA"
```

Output:



```
{
  "RefreshSchemasStatus": {
    "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWYGAYUVLKIB732KEVWA",
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "Status": "refreshing",
    "LastRefreshDate": 1590019949.103
  }
}
```

- Per API i dettagli, vedere [RefreshSchemas](#) in AWS CLI Command Reference.

## reload-tables

Il seguente esempio di codice mostra come utilizzare `reload-tables`.

### AWS CLI

Per aggiornare l'elenco delle tabelle disponibili in un endpoint

L'`reload-tables` esempio seguente ricarica l'elenco delle tabelle disponibili in un endpoint.

```
aws dms reload-tables \
  --replication-task-arn "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII" \
  --tables-to-reload "SchemaName=prodrep,TableName=ACCT_BAL"
```

Output:

```
{
  "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
}
```

- Per API i dettagli, vedere [ReloadTables](#) in AWS CLI Command Reference.

## remove-tags-from-resource

Il seguente esempio di codice mostra come utilizzare `remove-tags-from-resource`.

## AWS CLI

Per rimuovere i tag da un'istanza di replica

L'`remove-tags-from-resource` seguente rimuove i tag da un'istanza di replica.

```
aws dms remove-tags-from-resource \  
  --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE \  
  --tag-keys Environment Project
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Tagging Resources](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [RemoveTagsFromResource](#) in AWS CLI Command Reference.

## `start-replication-task-assessment`

Il seguente esempio di codice mostra come utilizzare `start-replication-task-assessment`.

## AWS CLI

Per avviare una valutazione delle attività

L'`start-replication-task-assessment` seguente avvia una valutazione dell'attività di replica.

```
aws dms start-replication-task-assessment \  
  --replication-task-arn arn:aws:dms:us-east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

Output:

```
{  
  "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": "testing",
    "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
    "ReplicationTaskCreationDate": 1590524772.505,
    "ReplicationTaskStartDate": 1590789988.677,
    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
  }
}

```

Per ulteriori informazioni, vedere [Creazione di un rapporto di valutazione delle attività](#) nella Guida per l'utente del AWS Database Migration Service.

- Per API i dettagli, vedere [StartReplicationTaskAssessment](#) in AWS CLI Command Reference.

## start-replication-task

Il seguente esempio di codice mostra come utilizzare `start-replication-task`.

### AWS CLI

Per avviare un'attività di replica

L'command-nameesempio seguente elenca i widget disponibili nel tuo AWS account.

```

aws dms start-replication-task \
  --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII \
  --start-replication-task-type reload-target

```

Output:

```

{
  "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": "starting",
    "ReplicationTaskCreationDate": 1590524772.505,
    "ReplicationTaskStartDate": 1590619805.212,
    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
  }
}

```

Per ulteriori informazioni, vedere [Working with AWS DMS Tasks](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [StartReplicationTask](#) in AWS CLI Command Reference.

## stop-replication-task

Il seguente esempio di codice mostra come utilizzare `stop-replication-task`.

### AWS CLI

Per interrompere un'operazione

L'`stop-replication-task` esempio seguente interrompe un'operazione.

```

aws dms stop-replication-task \
  --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, vedere [Working with AWS DMS Tasks](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [StopReplicationTask](#) in AWS CLI Command Reference.

## test-connection

Il seguente esempio di codice mostra come utilizzare `test-connection`.

### AWS CLI

Per testare una connessione a un endpoint

L'esempio seguente verifica se è possibile accedere a un endpoint da un'istanza di replica.

```

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:6GGI6YPWYGAYUVLKIB732KEVWA

```

Output:

```

{
  "Connection": {
    "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T30M70UB5NM2LCVZF7JPGJRNUE",
    "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWYGAYUVLKIB732KEVWA",
    "Status": "testing",
  }
}

```

```
    "EndpointIdentifier": "src-database-1",  
    "ReplicationInstanceIdentifier": "my-repl-instance"  
  }  
}
```

Per ulteriori informazioni, vedere [Creazione di endpoint di origine e destinazione](#) nella AWS Database Migration Service User Guide.

- Per API i dettagli, vedere [TestConnection](#) in AWS CLI Command Reference.

## Esempi di utilizzo di Amazon DocumentDB AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon DocumentDB. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come chiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **add-tags-to-resource**

Il seguente esempio di codice mostra come utilizzare `add-tags-to-resource`.

AWS CLI

Per aggiungere uno o più tag a una risorsa specificata

L'`add-tags-to-resource` esempio seguente aggiunge tre tag a `sample-cluster`. Un tag (`CropB`) ha un nome chiave ma nessun valore.

```
aws docdb add-tags-to-resource \  
  --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \  
  --tags Key=CropB,Value=
```

```
--tags Key="CropA",Value="Apple" Key="CropB" Key="CropC",Value="Corn"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Amazon DocumentDB Resources nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta Command [AddTagsToResource](#) Reference AWS CLI .

## **apply-pending-maintenance-action**

Il seguente esempio di codice mostra come utilizzare `apply-pending-maintenance-action`.

AWS CLI

Fare in modo che le azioni di manutenzione in sospeso vengano eseguite durante la finestra di manutenzione successiva

L'`apply-pending-maintenance-action` seguente fa sì che tutte le azioni di aggiornamento del sistema vengano eseguite durante la successiva finestra di manutenzione programmata.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Applicazione degli aggiornamenti di Amazon DocumentDB](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ApplyPendingMaintenanceAction](#) Reference.

## **copy-db-cluster-parameter-group**

Il seguente esempio di codice mostra come utilizzare `copy-db-cluster-parameter-group`.

AWS CLI

Per duplicare un gruppo di parametri del cluster DB esistente

L'esempio seguente crea una copia del gruppo di parametri `custom-docdb3-6` denominato `custom-docdb3-6-copy`. Quando esegue la copia, aggiunge dei tag al nuovo gruppo di parametri.

```
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"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, [consulta Copiare un gruppo di parametri del cluster Amazon DocumentDB](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta Command [CopyDbClusterParameterGroup](#) Reference AWS CLI .

## copy-db-cluster-snapshot

Il seguente esempio di codice mostra come utilizzare `copy-db-cluster-snapshot`.

AWS CLI

Per creare una copia di un'istantanea

L'esempio seguente dell'`copy-db-cluster-snapshot` copia `sample-cluster-snapshot` e assegna il nome `sample-cluster-snapshot-copy`. La copia contiene tutti i tag dell'originale più un nuovo tag con il nome `CopyNumber` della chiave.

```
aws docdb copy-db-cluster-snapshot \  
  --source-db-cluster-snapshot-identifier sample-cluster-snapshot \  
  --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \  
  --tags Key="CopyNumber",Value="1" Key="Modifiable",Value="Yes"
```



```
--target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \  
--copy-tags \  
--tags Key="CopyNumber",Value="1"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [consulta Copiare uno snapshot del cluster](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta Command [CopyDbClusterSnapshotReference](#) AWS CLI .

## create-db-cluster-parameter-group

Il seguente esempio di codice mostra come utilizzare `create-db-cluster-parameter-group`.

AWS CLI

Per creare un gruppo di parametri del cluster Amazon DocumentDB

L'`create-db-cluster-parameter-group` seguente crea il gruppo di parametri del cluster DB `sample-parameter-group` utilizzando la `docdb3.6` famiglia.

```
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"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Creating an Amazon DocumentDB Cluster Parameter Group](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateDbClusterParameterGroupReference](#).

## create-db-cluster-snapshot

Il seguente esempio di codice mostra come utilizzare `create-db-cluster-snapshot`.

### AWS CLI

Per creare uno snapshot manuale del cluster Amazon DocumentDB

L'esempio seguente crea uno snapshot del cluster Amazon DB denominato `sample-cluster-snapshot`.

```
aws docdb create-db-cluster-snapshot \  
  --db-cluster-identifier sample-cluster \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Output:

```
{  
  "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"
    }
}

```

Per ulteriori informazioni, consulta [Creazione di uno snapshot manuale del cluster](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateDbClusterSnapshotReference](#).

## create-db-cluster

Il seguente esempio di codice mostra come utilizzare `create-db-cluster`.

### AWS CLI

Per creare un cluster Amazon DocumentDB

L'esempio seguente crea un cluster Amazon DocumentDB denominato `sample-cluster` con la finestra di manutenzione preferita la domenica tra le 20:30 e le 11:00.

```

aws docdb create-db-cluster \
  --db-cluster-identifier sample-cluster \
  --engine docdb \
  --master-username master-user \
  --master-user-password password \
  --preferred-maintenance-window Sun:20:30-Sun:21:00

```

Output:

```

{
  "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"
}
}
```

Per ulteriori informazioni, consulta [Creating an Amazon DocumentDB Cluster nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta AWS CLI Command [CreateDbCluster](#)Reference.

## create-db-instance

Il seguente esempio di codice mostra come utilizzare `create-db-instance`.

### AWS CLI

Per creare un'istanza del cluster Amazon DocumentDB

Il codice di `create-db-instance` esempio seguente crea l'istanza `sample-cluster-instance-2` nel cluster Amazon DocumentDB. `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
```

Output:

```
{  
  "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"
}
}

```

Per ulteriori informazioni, consulta [Aggiungere un'istanza Amazon DocumentDB a un cluster](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateDbInstance](#)Reference.

## create-db-subnet-group

Il seguente esempio di codice mostra come utilizzare `create-db-subnet-group`.

## AWS CLI

Per creare un sottogruppo di Amazon DocumentDB

L'create-db-subnet-groupesempio seguente crea un sottogruppo di Amazon DocumentDB denominato. sample-subnet-group

```
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"
```

Output:

```
{  
  "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"  
        }  
      }  
    ]  
  }  
}
```

```
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Creating an Amazon DocumentDB Subnet Group nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta Command [CreateDbSubnetGroup](#)Reference AWS CLI .

## **delete-db-cluster-parameter-group**

Il seguente esempio di codice mostra come utilizzare `delete-db-cluster-parameter-group`.

### AWS CLI

Per eliminare un gruppo di parametri del cluster Amazon DocumentDB

L'`delete-db-cluster-parameter-group`esempio seguente elimina il gruppo di parametri Amazon DocumentDB. `sample-parameter-group`

```
aws docdb delete-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name sample-parameter-group
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di un gruppo di parametri del cluster Amazon DocumentDB](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta Command [DeleteDbClusterParameterGroup](#)Reference AWS CLI .

## **delete-db-cluster-snapshot**

Il seguente esempio di codice mostra come utilizzare `delete-db-cluster-snapshot`.

### AWS CLI

Per eliminare uno snapshot del cluster Amazon DocumentDB

L'`delete-db-cluster-snapshot`esempio seguente elimina lo snapshot del cluster Amazon DocumentDB. `sample-cluster-snapshot`

```
aws docdb delete-db-cluster-snapshot \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```



```
--db-cluster-snapshot-identifier sample-cluster-snapshot
```

Output:

```
{
  "DBClusterSnapshot": {
    "DBClusterIdentifier": "sample-cluster",
    "AvailabilityZones": [
      "us-west-2a",
      "us-west-2b",
      "us-west-2c",
      "us-west-2d"
    ],
    "DBClusterSnapshotIdentifier": "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
  }
}
```

Per ulteriori informazioni, consulta [Eliminazione di uno snapshot del cluster](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta Command [DeleteDbClusterSnapshotReference](#) AWS CLI .

## delete-db-cluster

Il seguente esempio di codice mostra come utilizzare `delete-db-cluster`.

### AWS CLI

Per eliminare un cluster Amazon DocumentDB

L'`delete-db-cluster` seguente elimina il cluster Amazon `sample-cluster` DocumentDB. Non viene eseguito alcun backup del cluster prima di eliminarlo. NOTE: È necessario eliminare tutte le istanze associate al cluster prima di poterlo eliminare.

```
aws docdb delete-db-cluster \  
  --db-cluster-identifier sample-cluster \  
  --skip-final-snapshot
```

Output:

```
{  
  "DBCluster": {  
    "DBClusterIdentifier": "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": "ZNKXH85TT8WVW",
    "BackupRetentionPeriod": 1,
    "DBClusterMembers": []
  }
}

```

Per ulteriori informazioni, consulta [Eliminazione di un cluster Amazon DocumentDB nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta Command [DeleteDbCluster](#)Reference AWS CLI .

## delete-db-instance

Il seguente esempio di codice mostra come utilizzare `delete-db-instance`.

### AWS CLI

Per eliminare un'istanza di Amazon DocumentDB

L'`delete-db-instance`esempio seguente elimina l'istanza Amazon `sample-cluster-instance-2` DocumentDB.

```

aws docdb delete-db-instance \
  --db-instance-identifier sample-cluster-instance-2

```

Output:

```

{
  "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.corcjozrlsfc.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
}
}

```

Per ulteriori informazioni, consulta [Eliminazione di un'istanza di Amazon DocumentDB nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta Command [DeleteDbInstance](#) Reference AWS CLI .

## delete-db-subnet-group

Il seguente esempio di codice mostra come utilizzare `delete-db-subnet-group`.

### AWS CLI

Per eliminare un sottogruppo di Amazon DocumentDB

L'`delete-db-subnet-group` seguente elimina il gruppo di sottoreti Amazon DocumentDB. `sample-subnet-group`

```

aws docdb delete-db-subnet-group \
  --db-subnet-group-name sample-subnet-group

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di un gruppo di sottoreti Amazon DocumentDB nella Amazon DocumentDB Developer Guide](#).

- Per i API dettagli, consulta [DeleteDbSubnetGroup](#) Command Reference. AWS CLI

## describe-db-cluster-parameter-groups

Il seguente esempio di codice mostra come utilizzare `describe-db-cluster-parameter-groups`.

### AWS CLI

Per visualizzare i dettagli di uno o più gruppi di parametri del cluster Amazon DocumentDB

L'esempio seguente mostra i dettagli per il gruppo di parametri del cluster Amazon DocumentDB. `custom3-6-param-grp`

```
aws docdb describe-db-cluster-parameter-groups \
  --db-cluster-parameter-group-name custom3-6-param-grp
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione [Visualizzazione dei gruppi di parametri del cluster Amazon DocumentDB](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeDbClusterParameterGroups](#) Reference.

## describe-db-cluster-parameters

Il seguente esempio di codice mostra come utilizzare `describe-db-cluster-parameters`.

### AWS CLI

Per visualizzare l'elenco dettagliato dei parametri per un gruppo di parametri del cluster Amazon DocumentDB.

L'`describe-db-cluster-parameters` seguente elenca i parametri per il gruppo di parametri Amazon DocumentDB `custom3-6-param-grp`.

```
aws docdb describe-db-cluster-parameters \  
  --db-cluster-parameter-group-name custom3-6-param-grp
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

```
]
}
```

Per ulteriori informazioni, consulta la sezione [Visualizzazione dei parametri del cluster Amazon DocumentDB](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeDbClusterParameters](#) Reference.

## describe-db-cluster-snapshot-attributes

Il seguente esempio di codice mostra come utilizzare `describe-db-cluster-snapshot-attributes`.

### AWS CLI

Per elencare i nomi e i valori degli attributi di uno snapshot di Amazon DocumentDB

L'`describe-db-cluster-snapshot-attributes` esempio seguente elenca i nomi e i valori degli attributi per lo snapshot di Amazon DocumentDB. `sample-cluster-snapshot`

```
aws docdb describe-db-cluster-snapshot-attributes \
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Output:

```
{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": []
      }
    ],
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
  }
}
```

Per ulteriori informazioni, consulta [DescribeDBCluster SnapshotAttributes](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta [DescribeDbClusterSnapshotAttributes AWS CLI Command Reference](#).



## describe-db-cluster-snapshots

Il seguente esempio di codice mostra come utilizzare `describe-db-cluster-snapshots`.

### AWS CLI

Per descrivere le istantanee di Amazon DocumentDB

L'esempio seguente mostra i dettagli per lo snapshot di Amazon DocumentDB. `sample-cluster-snapshot`

```
aws docdb describe-db-cluster-snapshots \
  --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [D escribeDBCluster Snapshots](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeDbClusterSnapshots](#)Reference.

## describe-db-clusters

Il seguente esempio di codice mostra come utilizzare `describe-db-clusters`.

### AWS CLI

Per ottenere informazioni dettagliate su uno o più cluster Amazon DocumentDB.

L'esempio seguente mostra i dettagli per il cluster Amazon DocumentDB. `sample-cluster` Omettendo il `--db-cluster-identifier` parametro è possibile ottenere informazioni su un massimo di 100 cluster.

```
aws docdb describe-db-clusters
--db-cluster-identifier sample-cluster
```

### Output:

```
{
  "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"
}
]
}

```

Per ulteriori informazioni, consulta [Describing Amazon DocumentDB Clusters nella Amazon DocumentDB Developer Guide](#).

- Per i API dettagli, consulta [DescribeDbClusters](#) Command Reference.AWS CLI

## describe-db-engine-versions

Il seguente esempio di codice mostra come utilizzare `describe-db-engine-versions`.

### AWS CLI

Per elencare le versioni disponibili del motore Amazon DocumentDB

L'`describe-db-engine-versions` seguente elenca tutte le versioni del motore Amazon DocumentDB disponibili.

```
aws docdb describe-db-engine-versions \  
  --engine docdb
```

Output:

```
{  
  "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"  
      ]  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [DescribeDBEngine Versions](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta [DescribeDbEngineVersions AWS CLI Command Reference](#).

## describe-db-instances

Il seguente esempio di codice mostra come utilizzare `describe-db-instances`.

## AWS CLI

Per trovare informazioni sulle istanze Amazon DocumentDB fornite

L'`describe-db-instances` seguente mostra i dettagli sull'istanza di Amazon DocumentDB. `sample-cluster-instance` Omettendo il `--db-instance-identifier` parametro si ottengono informazioni su un massimo di 100 istanze.

```
aws docdb describe-db-instances \
  --db-instance-identifier sample-cluster-instance
```

Output:

```
{
  "DBInstances": [
    {
      "Endpoint": {
        "HostedZoneId": "ZNKXH85TT8WVW",
        "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"
}

```

```
]
}
```

Per ulteriori informazioni, consulta [Describing Amazon DocumentDB Instances nella Amazon DocumentDB Developer Guide](#).

- Per i API dettagli, consulta [DescribeDbInstances](#) Command Reference.AWS CLI

## describe-db-subnet-groups

Il seguente esempio di codice mostra come utilizzare `describe-db-subnet-groups`.

### AWS CLI

Per recuperare un elenco di descrizioni delle sottoreti di Amazon DocumentDB

L'`describe-db-subnet-groups`esempio seguente descrive i dettagli per la sottorete Amazon DocumentDB denominata. `default`

```
aws docdb describe-db-subnet-groups \
  --db-subnet-group-name default
```

Output:

```
{
  "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"
}
]
}

```

Per ulteriori informazioni, consulta [Describing Subnet Groups](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta Command [DescribeDbSubnetGroups](#)Reference AWS CLI .

## **describe-engine-default-cluster-parameters**

Il seguente esempio di codice mostra come utilizzare `describe-engine-default-cluster-parameters`.

### AWS CLI

Per descrivere le informazioni predefinite sul motore e sui parametri di sistema per Amazon DocumentDB

L'`describe-engine-default-cluster-parameters` esempio seguente mostra i dettagli per il motore predefinito e le informazioni sui parametri di sistema per il gruppo di parametri Amazon DocumentDB. `docdb3.6`



```
aws docdb describe-engine-default-cluster-parameters \  
--db-parameter-group-family docdb3.6
```

Output:

```
{  
  "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  
      }  
    ]  
  }  
}
```

```
}  
}
```

Per ulteriori informazioni, consulta [DescribeEngineDefaultClusterParameters](#) la Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta [DescribeEngineDefaultClusterParameters AWS CLI Command Reference](#).

## describe-event-categories

Il seguente esempio di codice mostra come utilizzare `describe-event-categories`.

### AWS CLI

Per descrivere tutte le categorie di eventi di Amazon DocumentDB

L'esempio seguente elenca tutte le categorie per il tipo di sorgente di eventi Amazon DocumentDB. `db-instance`

```
aws docdb describe-event-categories \  
  --source-type db-cluster
```

Output:

```
{  
  "EventCategoriesMapList": [  
    {  
      "SourceType": "db-cluster",  
      "EventCategories": [  
        "failover",  
        "maintenance",  
        "notification",  
        "failure"  
      ]  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Viewing Event Categories](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta [DescribeEventCategories AWS CLI Command Reference](#).

## describe-events

Il seguente esempio di codice mostra come utilizzare `describe-events`.

### AWS CLI

Per elencare gli eventi di Amazon DocumentDB

L'`describe-events` esempio seguente elenca tutti gli eventi di Amazon DocumentDB delle ultime 24 ore (1440 minuti).

```
aws docdb describe-events \  
  --duration 1440
```

Questo comando non produce alcun output. Output:

```
{  
  "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"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Viewing Amazon DocumentDB Events nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta AWS CLI Command [DescribeEvents](#)Reference.

## describe-orderable-db-instance-options

Il seguente esempio di codice mostra come utilizzare `describe-orderable-db-instance-options`.

## AWS CLI

Per trovare le opzioni di istanza di Amazon DocumentDB che puoi ordinare

L'output di `describe-orderable-db-instance-options` seguente elenca tutte le opzioni di istanza per Amazon DocumentDB per una regione.

```
aws docdb describe-orderable-db-instance-options \  
  --engine docdb \  
  --region us-east-1
```

Output:

```
{  
  "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.r4.16xlarge",  
      "LicenseModel": "na",  
      "Engine": "docdb"  
    }  
  ]  
}
```

```
        "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"
  }
]
```

Per ulteriori informazioni, consulta [Aggiungere un'istanza Amazon DocumentDB a un cluster](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeOrderableDbInstanceOptions](#) Reference.

## describe-pending-maintenance-actions

Il seguente esempio di codice mostra come utilizzare `describe-pending-maintenance-actions`.

### AWS CLI

Per elencare le azioni di manutenzione di Amazon DocumentDB in sospeso

L'esempio seguente elenca tutte le azioni di manutenzione in sospeso di Amazon DocumentDB.

```
aws docdb describe-pending-maintenance-actions
```

Output:

```
{
  "PendingMaintenanceActions": []
}
```

Per ulteriori informazioni, consulta Maintenance [Amazon DocumentDB nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta [AWS CLI Command DescribePendingMaintenanceActionsReference](#).

## failover-db-cluster

Il seguente esempio di codice mostra come utilizzare `failover-db-cluster`.

### AWS CLI

Per forzare il failover di un cluster Amazon DocumentDB su una replica

L'`failover-db-cluster` esempio seguente fa sì che l'istanza primaria nel cluster Amazon DocumentDB `sample-cluster` esegua il failover su una replica.

```
aws docdb failover-db-cluster \  
--db-cluster-identifier sample-cluster
```

Output:

```
{  
  "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": "ZNKXH85TT8WVW",
  "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"
}
}

```

Per ulteriori informazioni, consulta [Amazon DocumentDB Failover nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta Command [FailoverDbCluster](#) Reference AWS CLI .

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

## AWS CLI

Per elencare tutti i tag su una risorsa Amazon DocumentDB

L'`list-tags-for-resource` seguente elenca tutti i tag sul cluster Amazon DocumentDB. `sample-cluster`

```
aws docdb list-tags-for-resource \  
  --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster
```

Output:

```
{  
  "TagList": [  
    {  
      "Key": "A",  
      "Value": "ALPHA"  
    },  
    {  
      "Key": "B",  
      "Value": ""  
    },  
    {  
      "Key": "C",  
      "Value": "CHARLIE"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Listing Tags on an Amazon DocumentDB Resource nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta AWS CLI Command [ListTagsForResource](#) Reference.

## `modify-db-cluster-parameter-group`

Il seguente esempio di codice mostra come utilizzare `modify-db-cluster-parameter-group`.

## AWS CLI

Per modificare un gruppo di parametri del cluster Amazon DocumentDB DB

L'`modify-db-cluster-parameter-group` seguente modifica il `custom3-6-param-grp` gruppo di parametri del cluster Amazon DocumentDB impostando i due `audit_logs` parametri `t1_monitor` e attivandolo. Le modifiche vengono applicate al riavvio successivo.

```
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=t1_monitor,ParameterValue=enabled,ApplyMethod=pending-reboot
```

Output:

```
{  
  "DBClusterParameterGroupName": "custom3-6-param-grp"  
}
```

Per ulteriori informazioni, consulta [Modificare un gruppo di parametri del cluster Amazon DocumentDB](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta Command [ModifyDbClusterParameterGroup](#) Reference AWS CLI .

## `modify-db-cluster-snapshot-attribute`

Il seguente esempio di codice mostra come utilizzare `modify-db-cluster-snapshot-attribute`.

AWS CLI

Esempio 1: aggiungere un attributo a uno snapshot di Amazon DocumentDB

L'`modify-db-cluster-snapshot-attribute` seguente aggiunge quattro valori di attributo a uno snapshot del cluster Amazon DocumentDB.

```
aws docdb modify-db-cluster-snapshot-attribute \  
  --db-cluster-snapshot-identifier sample-cluster-snapshot \  
  --attribute-name restore \  
  --values-to-add 123456789011 123456789012 123456789013
```

Output:

```
{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123456789011",
          "123456789012",
          "123456789013"
        ]
      }
    ],
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
  }
}
```

Esempio 2: rimuovere gli attributi da uno snapshot di Amazon DocumentDB

L'`modify-db-cluster-snapshot-attribute` seguente rimuove due valori di attributo da uno snapshot del cluster Amazon DocumentDB.

```
aws docdb modify-db-cluster-snapshot-attribute \
  --db-cluster-snapshot-identifier sample-cluster-snapshot \
  --attribute-name restore \
  --values-to-remove 123456789012
```

Output:

```
{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123456789011",
          "123456789013"
        ]
      }
    ],
    "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
  }
}
```

Per ulteriori informazioni, consulta [ModifyDBCluster SnapshotAttribute](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta [ModifyDbClusterSnapshotAttribute AWS CLI Command Reference](#).

## modify-db-cluster

Il seguente esempio di codice mostra come utilizzare `modify-db-cluster`.

### AWS CLI

Per modificare un cluster Amazon DocumentDB

L'`modify-db-cluster` seguente modifica il `sample-cluster` cluster Amazon DocumentDB impostando il periodo di conservazione per i backup automatici di 7 giorni e cambiando le finestre preferite per i backup e la manutenzione. Tutte le modifiche vengono applicate nella finestra di manutenzione successiva.

```
aws docdb modify-db-cluster \
  --db-cluster-identifier 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
```

Output:

```
{
  "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
}
}

```

Per ulteriori informazioni, consulta [Modifying an Amazon DocumentDB Cluster nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta Command [ModifyDbClusterReference](#) AWS CLI .

## modify-db-instance

Il seguente esempio di codice mostra come utilizzare `modify-db-instance`.

### AWS CLI

Per modificare un'istanza di Amazon DocumentDB

L'esempio seguente modifica l'istanza `sample-cluster2` di Amazon DocumentDB cambiando la classe di istanza in `db.r4.4xlarge` e il livello di promozione in `5`. Le modifiche vengono applicate immediatamente, ma possono essere visualizzate solo dopo che lo stato dell'istanza è disponibile.

```
aws docdb modify-db-instance \  
  --db-instance-identifier sample-cluster2 \  
  --apply-immediately \  
  --db-instance-class db.r4.4xlarge \  
  --promotion-tier 5
```

Output:

```
{  
  "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"
      },
      "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"
      },
      "SubnetStatus": "Active"
    }
  ],
  "VpcId": "vpc-91280df6"
},
"PromotionTier": 2,
"Endpoint": {
  "Address": "sample-cluster2.corcjozrlsfc.us-west-2.docdb.amazonaws.com",
  "HostedZoneId": "ZNKXH85TT8WVW",
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Modificare un'istanza di Amazon DocumentDB nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta Command [ModifyDbInstance](#) Reference AWS CLI .

## modify-db-subnet-group

Il seguente esempio di codice mostra come utilizzare `modify-db-subnet-group`.

### AWS CLI

Per modificare un sottogruppo di Amazon DocumentDB

L'`modify-db-subnet-group` esempio seguente modifica il gruppo di sottoreti `sample-subnet-group` aggiungendo le sottoreti specificate e una nuova descrizione.

```

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"

```

Output:

```

{
  "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"
    }
  },
  {
    "SubnetIdentifier": "subnet-991cb8d0",
    "SubnetStatus": "Active",
    "SubnetAvailabilityZone": {
      "Name": "us-west-2b"
    }
  }
]
}
```

Per ulteriori informazioni, consulta [Modifying an Amazon DocumentDB Subnet Group nella Amazon DocumentDB Developer Guide](#).

- Per i API dettagli, consulta [ModifyDbSubnetGroup](#) Command Reference.AWS CLI

## reboot-db-instance

Il seguente esempio di codice mostra come utilizzare `reboot-db-instance`.

### AWS CLI

Per riavviare un'istanza di Amazon DocumentDB

L'`reboot-db-instance`esempio seguente riavvia l'istanza Amazon `sample-cluster2` DocumentDB.

```
aws docdb reboot-db-instance \  
--db-instance-identifier sample-cluster2
```

Questo comando non produce alcun output. Output:

```
{  
  "DBInstance": {  
    "PreferredBackupWindow": "18:00-18:30",  
    "DBInstanceIdentifier": "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"
}
}

```

Per ulteriori informazioni, consulta [Rebooting an Amazon DocumentDB nella Amazon Instance Developer Guide](#).

- Per i API dettagli, consulta [RebootDbInstance](#) Command Reference.AWS CLI

## remove-tags-from-resource

Il seguente esempio di codice mostra come utilizzare `remove-tags-from-resource`.

## AWS CLI

Per rimuovere tag da una risorsa Amazon DocumentDB

L'`remove-tags-from-resource` esempio seguente rimuove il tag con la chiave denominata `B` dal cluster Amazon DocumentDB. `sample-cluster`

```
aws docdb remove-tags-from-resource \  
  --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \  
  --tag-keys B
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Removing tags from an Amazon DocumentDB Resource](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta [RemoveTagsFromResource AWS CLI Command Reference](#).

## `reset-db-cluster-parameter-group`

Il seguente esempio di codice mostra come utilizzare `reset-db-cluster-parameter-group`.

## AWS CLI

Per ripristinare il valore del parametro specificato ai valori predefiniti in un gruppo di parametri Amazon DocumentDB

L'`reset-db-cluster-parameter-group` esempio seguente reimposta il parametro `ttl_monitor` nel `custom3-6-param-grp` gruppo di parametri Amazon DocumentDB al valore predefinito.

```
aws docdb reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --parameters ParameterName=ttl_monitor,ApplyMethod=immediate
```

Output:

```
{  
  "DBClusterParameterGroupName": "custom3-6-param-grp"  
}
```

Per ulteriori informazioni, consulta il titolo della Amazon DocumentDB Developer Guide.



Per ripristinare i valori predefiniti di alcuni parametri o di tutti i valori predefiniti in un gruppo di parametri Amazon DocumentDB

L'`reset-db-cluster-parameter-group` seguente reimposta tutti i parametri nel `custom3-6-param-grp` gruppo di parametri Amazon DocumentDB al loro valore predefinito.

```
aws docdb reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name custom3-6-param-grp \  
  --reset-all-parameters
```

Output:

```
{  
  "DBClusterParameterGroupName": "custom3-6-param-grp"  
}
```

Per ulteriori informazioni, consulta [Reimpostazione di un gruppo di parametri del cluster Amazon DocumentDB nella Amazon DocumentDB Developer Guide](#).

- Per i API dettagli, consulta [ResetDbClusterParameterGroup](#) Command Reference.AWS CLI

## restore-db-cluster-from-snapshot

Il seguente esempio di codice mostra come utilizzare `restore-db-cluster-from-snapshot`.

AWS CLI

Per ripristinare un cluster Amazon DocumentDB da uno snapshot automatico o manuale

L'`restore-db-cluster-from-snapshot` seguente crea un nuovo cluster Amazon DocumentDB denominato in base allo `sample-cluster-2019-03-16-00-01-restored` snapshot. `rds:sample-cluster-2019-03-16-00-01`

```
aws docdb restore-db-cluster-from-snapshot \  
  --db-cluster-identifier sample-cluster-2019-03-16-00-01-restored \  
  --engine docdb \  
  --snapshot-identifier rds:sample-cluster-2019-03-16-00-01
```

Output:

```
{
```

```

"DBCluster": {
  "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
  "HostedZoneId": "ZNKXH85TT8WVW",
  "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-
corcjorzrlsfc.us-west-2.docdb.amazonaws.com",
  "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjorzrlsfc.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"
    }
  ]
}
}

```

Per ulteriori informazioni, consulta [Restoring from a Cluster Snapshot](#) nella Amazon DocumentDB Developer Guide.

- Per API i dettagli, consulta Command [RestoreDbClusterFromSnapshot](#)Reference AWS CLI .

## restore-db-cluster-to-point-in-time

Il seguente esempio di codice mostra come utilizzare `restore-db-cluster-to-point-in-time`.

### AWS CLI

Per ripristinare un cluster Amazon DocumentDB point-in-time da uno snapshot manuale

L'esempio seguente utilizza `sample-cluster-snapshot` per creare un nuovo cluster Amazon DocumentDB `sample-cluster-pit`, utilizzando l'ora di ripristino più recente.

```
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
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Restoring a snapshot to a point in time nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta Command [RestoreDbClusterToPointInTime](#) Reference AWS CLI .

## start-db-cluster

Il seguente esempio di codice mostra come utilizzare `start-db-cluster`.

### AWS CLI

Per avviare un cluster Amazon DocumentDB interrotto

L'`start-db-cluster` esempio seguente avvia il cluster Amazon DocumentDB specificato.

```

aws docdb start-db-cluster \
  --db-cluster-identifier sample-cluster

```

Output:

```

{
  "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"
        }
    ]
}
}

```

Per ulteriori informazioni, consulta [Stopping and Starting an Amazon DocumentDB Cluster nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta Command [StartDbCluster](#)Reference AWS CLI .

## stop-db-cluster

Il seguente esempio di codice mostra come utilizzare `stop-db-cluster`.

## AWS CLI

Per arrestare un cluster Amazon DocumentDB in esecuzione

L'`stop-db-cluster` seguente arresta il cluster Amazon DocumentDB specificato.

```
aws docdb stop-db-cluster \  
--db-cluster-identifier sample-cluster
```

Output:

```
{  
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Stopping and Starting an Amazon DocumentDB Cluster nella Amazon DocumentDB Developer Guide](#).

- Per API i dettagli, consulta Command [StopDbCluster](#)Reference AWS CLI .

## Esempi di utilizzo di DynamoDB AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface con DynamoDB.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **batch-get-item**

Il seguente esempio di codice mostra come utilizzare `batch-get-item`.

AWS CLI

Per recuperare più elementi da una tabella

L'esempio seguente legge più elementi dalla `MusicCollection` tabella utilizzando un batch di tre `GetItem` richieste e richiede il numero di unità di capacità di lettura utilizzate dall'operazione. Il comando restituisce solo l'`AlbumTitle` attributo.

```
aws dynamodb batch-get-item \  
  --request-items file://request-items.json \  
  --return-consumed-capacity TOTAL
```

Contenuto di `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"  
  }  
}
```

Output:

```
{  
  "Responses": {  
    "MusicCollection": [  
      {  
        "AlbumTitle": {  
          "S": "Somewhat Famous"  
        }  
      },  
      {  
        "AlbumTitle": {  
          "S": "Blue Sky Blues"  
        }  
      },  
      {
```



```

        "AlbumTitle": {
            "S": "Louder Than Ever"
        }
    ]
},
"UnprocessedKeys": {},
"ConsumedCapacity": [
    {
        "TableName": "MusicCollection",
        "CapacityUnits": 1.5
    }
]
}

```

Per ulteriori informazioni, consulta [Batch Operations](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [BatchGetItem](#) Reference.

## batch-write-item

Il seguente esempio di codice mostra come utilizzare `batch-write-item`.

### AWS CLI

Per aggiungere più elementi a una tabella

L'esempio seguente aggiunge tre nuovi elementi alla `MusicCollection` tabella utilizzando un batch di tre `PutItem` richieste. Richiede inoltre informazioni sul numero di unità di capacità di scrittura utilizzate dall'operazione e sulle eventuali raccolte di elementi modificate dall'operazione.

```

aws dynamodb batch-write-item \
  --request-items file://request-items.json \
  --return-consumed-capacity INDEXES \
  --return-item-collection-metrics SIZE

```

Contenuto di `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"}
          }
        }
      }
    ]
  }
}

```

### Output:

```

{
  "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
      }
    }
  }
]
}

```

Per ulteriori informazioni, consulta [Batch Operations](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [BatchWriteItem](#) Reference.

## create-backup

Il seguente esempio di codice mostra come utilizzare `create-backup`.

### AWS CLI

Per creare un backup per una tabella DynamoDB esistente

L'`create-backup` seguente crea un backup della `MusicCollection` tabella.

```
aws dynamodb create-backup \  
  --table-name MusicCollection \  
  --backup-name MusicCollectionBackup
```

Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, consulta [Backup e ripristino su richiesta per DynamoDB nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [CreateBackup](#) AWS CLI

## `create-global-table`

Il seguente esempio di codice mostra come utilizzare `create-global-table`.

AWS CLI

Per creare una tabella globale

L'`create-global-table` seguente crea una tabella globale da due tabelle identiche nelle AWS regioni separate specificate.

```
aws dynamodb create-global-table \  
  --global-table-name MusicCollection \  
  --replication-group RegionName=us-east-2 RegionName=us-east-1 \  
  --region us-east-2
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [DynamoDB Global Tables nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta [CreateGlobalTable](#) Command Reference.AWS CLI

## create-table

Il seguente esempio di codice mostra come utilizzare `create-table`.

### AWS CLI

Esempio 1: creare una tabella con tag

L'`create-table`esempio seguente utilizza gli attributi e lo schema chiave specificati per creare una tabella denominata `MusicCollection`. Questa tabella utilizza la velocità effettiva assegnata ed è crittografata a riposo utilizzando la velocità di trasmissione predefinita AWS . CMK Il comando applica anche un tag alla tabella, con una chiave di `Owner` e un valore di `blueTeam`

```
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
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Basic Operations for Tables](#) nella Amazon DynamoDB Developer Guide.

## Esempio 2: creare una tabella in modalità On-Demand

L'esempio seguente crea una tabella chiamata `MusicCollection` utilizzando la modalità on-demand, anziché la modalità throughput assegnata. Questa funzionalità è utile per le tabelle con carichi di lavoro imprevedibili.

```
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
```

Output:

```
{  
  "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"
    }
  }
}

```

Per ulteriori informazioni, consulta [Basic Operations for Tables](#) nella Amazon DynamoDB Developer Guide.

Esempio 3: creare una tabella e crittografarla con un Customer Managed CMK

L'esempio seguente crea una tabella denominata `MusicCollection` e la crittografa utilizzando una tabella gestita dal cliente. 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

```

Output:

```

{
  "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"
}
}
}

```

Per ulteriori informazioni, consulta [Basic Operations for Tables](#) nella Amazon DynamoDB Developer Guide.

Esempio 4: creare una tabella con un indice secondario locale

L'esempio seguente utilizza gli attributi e lo schema chiave specificati per creare una tabella denominata `MusicCollection` con un indice secondario locale denominato `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\"]
        }
      }
    ]"
```

Output:

```
{
  "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"
    }
]
}
}

```

Per ulteriori informazioni, consulta [Basic Operations for Tables](#) nella Amazon DynamoDB Developer Guide.

Esempio 5: creare una tabella con un indice secondario globale

L'esempio seguente crea una tabella denominata `GameScores` con un indice secondario globale chiamato `GameTitleIndex`. La tabella base ha una chiave di partizione `UserId` e una chiave di ordinamento `GameTitle`, che consentono di trovare il punteggio migliore di un singolo utente per un gioco specifico in modo efficiente, mentre GSI ha una chiave di partizione `GameTitle` e una chiave di ordinamento `TopScore`, che consentono di trovare rapidamente il punteggio complessivo più alto per un determinato gioco.

```

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 \
    "[
      {
        \bIndexName\b": \b"GameTitleIndex\b",
        \bKeySchema\b": [
          {\bAttributeName\b":\b"GameTitle\b",\bKeyType\b":\b"HASH\b"},
          {\bAttributeName\b":\b"TopScore\b",\bKeyType\b":\b"RANGE\b"}
        ],
        \bProjection\b": {
          \bProjectionType\b":\b"INCLUDE\b",
          \bNonKeyAttributes\b":[\b"UserId\b"]
        },
        \bProvisionedThroughput\b": {

```

```

        \ "ReadCapacityUnits\": 10,
        \ "WriteCapacityUnits\": 5
    }
}
]"

```

## Output:

```

{
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Basic Operations for Tables](#) nella Amazon DynamoDB Developer Guide.

Esempio 6: creare una tabella con più indici secondari globali contemporaneamente

L'esempio seguente crea una tabella denominata `GameScores` con due indici secondari globali. Gli GSI schemi vengono passati tramite un file, anziché sulla riga di comando.

```
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
```

Contenuto di `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
  }
}
]
```

Output:

```
{
  "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"
  }
]
}
}

```

Per ulteriori informazioni, consulta [Basic Operations for Tables](#) nella Amazon DynamoDB Developer Guide.

### Esempio 7: creare una tabella con Streams abilitato

L'esempio seguente crea una tabella chiamata GameScores con DynamoDB Streams abilitato. Sia le immagini nuove che quelle vecchie di ogni elemento verranno scritte nello stream.

```

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
```

Output:

```
{  
  "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"
}
}

```

Per ulteriori informazioni, consulta [Basic Operations for Tables](#) nella Amazon DynamoDB Developer Guide.

Esempio 8: creare una tabella con Keys-Only Stream abilitato

L'esempio seguente crea una tabella chiamata GameScores con DynamoDB Streams abilitato. Nel flusso vengono scritti solo gli attributi chiave degli elementi modificati.

```

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

```

Output:

```

{
  "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
}
}

```

Per ulteriori informazioni, consulta [Change data capture for DynamoDB Streams nella Amazon DynamoDB Developer Guide](#).

Esempio 9: creare una tabella con la classe Standard Infrequent Access

L'esempio seguente crea una tabella chiamata `GameScores` e assegna la classe di tabella Standard-Infrequent Access (DynamoDB Standard-IA). Questa classe di tabelle è ottimizzata perché lo storage è il costo principale.

```

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

```

Output:

```

{
  "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
  }
}

```

Per ulteriori informazioni, consulta [Table classes](#) nella Amazon DynamoDB Developer Guide.

Esempio 10: creare una tabella con la protezione da eliminazione abilitata

L'esempio seguente crea una tabella denominata GameScores e abilita la protezione da eliminazione.

```

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

```

Output:

```

{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Using Deletion Protection](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateTable](#) Reference.

## delete-backup

Il seguente esempio di codice mostra come utilizzare `delete-backup`.

### AWS CLI

Per eliminare un backup DynamoDB esistente

L'`delete-backup` esempio seguente elimina il backup esistente specificato.

```
aws dynamodb delete-backup \  
  --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/  
backup/01576616366715-b4e58d3a
```

Output:



```

{
  "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": {}
  }
}

```

Per ulteriori informazioni, consulta [Backup e ripristino su richiesta per DynamoDB nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [DeleteBackup](#) AWS CLI

## delete-item

Il seguente esempio di codice mostra come utilizzare `delete-item`.

### AWS CLI

Esempio 1: Per eliminare un elemento

L'esempio seguente elimina un elemento dalla `MusicCollection` tabella e richiede i dettagli sull'elemento che è stato eliminato e sulla capacità utilizzata dalla richiesta.

```
aws dynamodb delete-item \  
  --table-name MusicCollection \  
  --key file://key.json \  
  --return-values ALL_OLD \  
  --return-consumed-capacity TOTAL \  
  --return-item-collection-metrics SIZE
```

Contenuto di `key.json`.

```
{  
  "Artist": {"S": "No One You Know"},  
  "SongTitle": {"S": "Scared of My Shadow"}  
}
```

Output:

```
{  
  "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
    ]
  }
}

```

Per ulteriori informazioni, consulta [Writing an Item](#) in Amazon DynamoDB Developer Guide.

Esempio 2: eliminare un elemento in modo condizionale

L'esempio seguente elimina un articolo dalla ProductCatalog tabella solo se ProductCategory è uno Sporting Goods o l'altro Gardening Supplies e il suo prezzo è compreso tra 500 e 600. Restituisce i dettagli sull'elemento che è stato eliminato.

```

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

```

Contenuto di names.json.

```

{
  "#P": "Price"
}

```

Contenuto di values.json.

```

{
  ":cat1": {"S": "Sporting Goods"},
  ":cat2": {"S": "Gardening Supplies"},
}

```

```
    ":lo": {"N": "500"},
    ":hi": {"N": "600"}
}
```

Output:

```
{
  "Attributes": {
    "Id": {
      "N": "456"
    },
    "Price": {
      "N": "550"
    },
    "ProductCategory": {
      "S": "Sporting Goods"
    }
  }
}
```

Per ulteriori informazioni, consulta [Writing an Item](#) in Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteItem](#) Reference.

## delete-table

Il seguente esempio di codice mostra come utilizzare `delete-table`.

AWS CLI

Per eliminare una tabella

L'esempio seguente elimina la `MusicCollection` tabella.

```
aws dynamodb delete-table \
  --table-name MusicCollection
```

Output:

```
{
  "TableDescription": {
    "TableStatus": "DELETING",
```

```

    "TableSizeBytes": 0,
    "ItemCount": 0,
    "TableName": "MusicCollection",
    "ProvisionedThroughput": {
      "NumberOfDecreasesToday": 0,
      "WriteCapacityUnits": 5,
      "ReadCapacityUnits": 5
    }
  }
}

```

Per ulteriori informazioni, consulta [Eliminazione di una tabella](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [DeleteTable](#)Reference AWS CLI .

## describe-backup

Il seguente esempio di codice mostra come utilizzare `describe-backup`.

### AWS CLI

Per ottenere informazioni su un backup esistente di una tabella

L'`describe-backup` esempio seguente visualizza informazioni sul backup esistente specificato.

```

aws dynamodb describe-backup \
  --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a

```

Output:

```

{
  "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": {}
  }
}
```

Per ulteriori informazioni, consulta [Backup e ripristino su richiesta per DynamoDB nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [DescribeBackup](#) AWS CLI

## describe-continuous-backups

Il seguente esempio di codice mostra come utilizzare `describe-continuous-backups`.

### AWS CLI

Per ottenere informazioni sui backup continui per una tabella DynamoDB

L'esempio seguente mostra i dettagli sulle impostazioni di backup continuo per la tabella `MusicCollection`

```
aws dynamodb describe-continuous-backups \  
  --table-name MusicCollection
```

Output:

```
{  
  "ContinuousBackupsDescription": {  
    "ContinuousBackupsStatus": "ENABLED",  
    "PointInTimeRecoveryDescription": {  
      "PointInTimeRecoveryStatus": "DISABLED"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Point-in-Time Recovery for DynamoDB](#) nella Amazon DynamoDB Developer Guide.

- Per i API dettagli, consulta [DescribeContinuousBackups](#) Command Reference.AWS CLI

## describe-contributor-insights

Il seguente esempio di codice mostra come utilizzare `describe-contributor-insights`.

AWS CLI

Per visualizzare le impostazioni di Contributor Insights per una tabella DynamoDB

L'esempio seguente visualizza le impostazioni di Contributor Insights per la `MusicCollection` tabella e l'`AlbumTitle-index` indice secondario globale.

```
aws dynamodb describe-contributor-insights \  
  --table-name MusicCollection \  
  --index-name AlbumTitle-index
```

Output:

```
{  
  "TableName": "MusicCollection",  
  "IndexName": "AlbumTitle-index",  
  "ContributorInsightsRuleList": [  
    {  
      "RuleName": "AlbumTitle-index",  
      "RuleType": "GlobalSecondaryIndex",  
      "Status": "ENABLED"  
    }  
  ]  
}
```

```
"DynamoDBContributorInsights-PKC-MusicCollection-1576629651520",
"DynamoDBContributorInsights-SKC-MusicCollection-1576629651520",
"DynamoDBContributorInsights-PKT-MusicCollection-1576629651520",
"DynamoDBContributorInsights-SKT-MusicCollection-1576629651520"
],
"ContributorInsightsStatus": "ENABLED",
"LastUpdateDateTime": 1576629654.78
}
```

Per ulteriori informazioni, consulta [Analyzing Data Access Using CloudWatch Contributor Insights for DynamoDB nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [DescribeContributorInsights](#)AWS CLI

## describe-endpoints

Il seguente esempio di codice mostra come utilizzare `describe-endpoints`.

AWS CLI

Per visualizzare le informazioni sugli endpoint regionali

L'esempio seguente mostra i dettagli sugli endpoint per la regione corrente AWS .

```
aws dynamodb describe-endpoints
```

Output:

```
{
  "Endpoints": [
    {
      "Address": "dynamodb.us-west-2.amazonaws.com",
      "CachePeriodInMinutes": 1440
    }
  ]
}
```

Per ulteriori informazioni, consulta [Amazon DynamoDB Endpoints and Quotas](#) nella Guida generale.AWS

- Per i API dettagli, consulta [DescribeEndpoints](#)Command Reference.AWS CLI



## describe-global-table-settings

Il seguente esempio di codice mostra come utilizzare `describe-global-table-settings`.

### AWS CLI

Per ottenere informazioni sulle impostazioni di una tabella globale DynamoDB

L'esempio seguente mostra le impostazioni per la tabella `MusicCollection` globale.

```
aws dynamodb describe-global-table-settings \
  --global-table-name MusicCollection
```

Output:

```
{
  "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
      }
    }
  ]
}
```

```
]
}
```

Per ulteriori informazioni, consulta [DynamoDB Global Tables nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta [DescribeGlobalTableSettings](#) Command Reference.AWS CLI

## describe-global-table

Il seguente esempio di codice mostra come utilizzare `describe-global-table`.

### AWS CLI

Per visualizzare informazioni su una tabella globale DynamoDB

L'esempio seguente mostra i dettagli sulla tabella `MusicCollection` globale.

```
aws dynamodb describe-global-table \
  --global-table-name MusicCollection
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [DynamoDB Global Tables nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta [DescribeGlobalTable](#) Command Reference.AWS CLI

## **describe-limits**

Il seguente esempio di codice mostra come utilizzare `describe-limits`.

### AWS CLI

Per visualizzare i limiti di capacità assegnati

L'esempio seguente mostra i limiti di capacità prevista per il tuo account nella regione corrente. AWS

```
aws dynamodb describe-limits
```

Output:

```
{
  "AccountMaxReadCapacityUnits": 80000,
  "AccountMaxWriteCapacityUnits": 80000,
  "TableMaxReadCapacityUnits": 40000,
  "TableMaxWriteCapacityUnits": 40000
}
```

Per ulteriori informazioni, consulta [Limits in DynamoDB nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta [DescribeLimits](#) Command Reference.AWS CLI

## **describe-table-replica-auto-scaling**

Il seguente esempio di codice mostra come utilizzare `describe-table-replica-auto-scaling`.

### AWS CLI

Per visualizzare le impostazioni di ridimensionamento automatico tra le repliche di una tabella globale

L'output del comando `aws dynamodb describe-table-replica-auto-scaling` seguente visualizza le impostazioni di ridimensionamento automatico tra le repliche della tabella `MusicCollection` globale.

```
aws dynamodb describe-table-replica-auto-scaling \
  --table-name MusicCollection
```

Output:

```
{
  "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"
}
}

```

```
    ]
  }
}
```

Per ulteriori informazioni, consulta [DynamoDB Global Tables nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta [DescribeTableReplicaAutoScaling](#) Command Reference.AWS CLI

## describe-table

Il seguente esempio di codice mostra come utilizzare `describe-table`.

### AWS CLI

Per descrivere una tabella

L'`describe-table` esempio seguente descrive la `MusicCollection` tabella.

```
aws dynamodb describe-table \
  --table-name MusicCollection
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Describing a Table](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command [DescribeTable](#)Reference AWS CLI .

## describe-time-to-live

Il seguente esempio di codice mostra come utilizzare `describe-time-to-live`.

### AWS CLI

Per visualizzare le impostazioni Time to Live per una tabella

L'esempio seguente visualizza le impostazioni Time to Live per la `MusicCollection` tabella.

```
aws dynamodb describe-time-to-live \
  --table-name MusicCollection
```

Output:

```
{
  "TimeToLiveDescription": {
    "TimeToLiveStatus": "ENABLED",
    "AttributeName": "ttl"
  }
}
```

```
}
```

Per ulteriori informazioni, consulta [Time to Live](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeTimeToLiveReference](#).

## get-item

Il seguente esempio di codice mostra come utilizzare `get-item`.

### AWS CLI

Esempio 1: leggere un elemento in una tabella

L'esempio seguente recupera un elemento dalla `MusicCollection` tabella. La tabella ha una chiave hash-and-range primaria (`ArtistSongTitle`), quindi è necessario specificare entrambi questi attributi. Il comando richiede anche informazioni sulla capacità di lettura consumata dall'operazione.

```
aws dynamodb get-item \  
  --table-name MusicCollection \  
  --key file://key.json \  
  --return-consumed-capacity TOTAL
```

Contenuto di `key.json`.

```
{  
  "Artist": {"S": "Acme Band"},  
  "SongTitle": {"S": "Happy Day"}  
}
```

Output:

```
{  
  "Item": {  
    "AlbumTitle": {  
      "S": "Songs About Life"  
    },  
    "SongTitle": {  
      "S": "Happy Day"  
    },  
  },  
}
```



```
    "Artist": {
      "S": "Acme Band"
    },
    "ConsumedCapacity": {
      "TableName": "MusicCollection",
      "CapacityUnits": 0.5
    }
  }
}
```

Per ulteriori informazioni, consulta [Reading an Item](#) nella Amazon DynamoDB Developer Guide.

Esempio 2: leggere un elemento utilizzando una lettura coerente

L'esempio seguente recupera un elemento dalla MusicCollection tabella utilizzando letture fortemente coerenti.

```
aws dynamodb get-item \
  --table-name MusicCollection \
  --key file://key.json \
  --consistent-read \
  --return-consumed-capacity TOTAL
```

Contenuto di key.json.

```
{
  "Artist": {"S": "Acme Band"},
  "SongTitle": {"S": "Happy Day"}
}
```

Output:

```
{
  "Item": {
    "AlbumTitle": {
      "S": "Songs About Life"
    },
    "SongTitle": {
      "S": "Happy Day"
    },
    "Artist": {
      "S": "Acme Band"
    }
  }
}
```

```

    }
  },
  "ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 1.0
  }
}

```

Per ulteriori informazioni, consulta [Reading an Item](#) nella Amazon DynamoDB Developer Guide.

### Esempio 3: recuperare attributi specifici di un articolo

L'esempio seguente utilizza un'espressione di proiezione per recuperare solo tre attributi dell'elemento desiderato.

```

aws dynamodb get-item \
  --table-name ProductCatalog \
  --key '{"Id": {"N": "102"}}' \
  --projection-expression "#T, #C, #P" \
  --expression-attribute-names file://names.json

```

Contenuto di `names.json`.

```

{
  "#T": "Title",
  "#C": "ProductCategory",
  "#P": "Price"
}

```

Output:

```

{
  "Item": {
    "Price": {
      "N": "20"
    },
    "Title": {
      "S": "Book 102 Title"
    },
    "ProductCategory": {
      "S": "Book"
    }
  }
}

```

```
}  
}
```

Per ulteriori informazioni, consulta [Reading an Item](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetItem](#) Reference.

## list-backups

Il seguente esempio di codice mostra come utilizzare `list-backups`.

### AWS CLI

Esempio 1: Per elencare tutti i backup DynamoDB esistenti

L'`list-backups` seguente elenca tutti i backup esistenti.

```
aws dynamodb list-backups
```

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-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
    }
]
}

```

Per ulteriori informazioni, consulta [Backup e ripristino su richiesta per DynamoDB nella Amazon DynamoDB Developer Guide](#).

Esempio 2: per elencare i backup creati dall'utente in un intervallo di tempo specifico

L'esempio seguente elenca solo i backup della MusicCollection tabella creati dall'utente (non quelli creati automaticamente da DynamoDB) con una data di creazione compresa tra il 1° gennaio 2020 e il 1° marzo 2020.

```

aws dynamodb list-backups \
  --table-name MusicCollection \
  --time-range-lower-bound 1577836800 \
  --time-range-upper-bound 1583020800 \
  --backup-type USER

```

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-a1bcd234",
      "BackupName": "MusicCollectionBackup1",
      "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
      "BackupStatus": "AVAILABLE",
      "BackupType": "USER",
      "BackupSizeBytes": 170
    }
  ]
}

```

```
}
```

Per ulteriori informazioni, consulta [Backup e ripristino su richiesta per DynamoDB nella Amazon DynamoDB Developer Guide](#).

Esempio 3: limitare le dimensioni della pagina

L'esempio seguente restituisce un elenco di tutti i backup esistenti, ma recupera solo un elemento in ogni chiamata, eseguendo più chiamate se necessario per ottenere l'intero elenco. La limitazione delle dimensioni della pagina è utile quando si eseguono comandi di elenco su un numero elevato di risorse, il che può causare un errore di «timeout» quando si utilizza la dimensione di pagina predefinita di 1000.

```
aws dynamodb list-backups \  
  --page-size 1
```

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-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
    }
]
}

```

Per ulteriori informazioni, consulta [Backup e ripristino su richiesta per DynamoDB nella Amazon DynamoDB Developer Guide](#).

Esempio 4: limitare il numero di articoli restituiti

L'esempio seguente limita il numero di articoli restituiti a 1. La risposta include un NextToken valore con cui recuperare la pagina successiva di risultati.

```

aws dynamodb list-backups \
  --max-items 1

```

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-a1bcd234",
      "BackupName": "MusicCollectionBackup1",
      "BackupCreationDateTime": "2020-02-12T14:41:51.617000-08:00",
      "BackupStatus": "AVAILABLE",
      "BackupType": "USER",
      "BackupSizeBytes": 170
    }
  ],
  "NextToken":
  "abCDeFGhiJKlMnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}

```

Per ulteriori informazioni, consulta [Backup e ripristino su richiesta per DynamoDB nella Amazon DynamoDB Developer Guide](#).

### Esempio 5: per recuperare la pagina successiva dei risultati

Il comando seguente utilizza il `NextToken` valore di una precedente chiamata al `list-backups` comando per recuperare un'altra pagina di risultati. Poiché la risposta in questo caso non include un `NextToken` valore, sappiamo di aver raggiunto la fine dei risultati.

```
aws dynamodb list-backups \  
  --starting-  
  token abCDeFGhiJKLmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

### 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  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Backup e ripristino su richiesta per DynamoDB nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [ListBackups](#)AWS CLI

## list-contributor-insights

Il seguente esempio di codice mostra come utilizzare `list-contributor-insights`.

### AWS CLI

Esempio 1: per visualizzare un elenco di riepiloghi di Contributor Insights

L'`list-contributor-insights` seguente visualizza un elenco di riepiloghi di Contributor Insights.

```
aws dynamodb list-contributor-insights
```

Output:

```
{
  "ContributorInsightsSummaries": [
    {
      "TableName": "MusicCollection",
      "IndexName": "AlbumTitle-index",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "ProductCatalog",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Forum",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Reply",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Thread",
      "ContributorInsightsStatus": "ENABLED"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Analyzing Data Access Using CloudWatch Contributor Insights for DynamoDB nella Amazon DynamoDB Developer Guide](#).

Esempio 2: limitare il numero di articoli restituiti

L'esempio seguente limita il numero di articoli restituiti a 4. La risposta include un `NextToken` valore con cui recuperare la pagina successiva di risultati.

```
aws dynamodb list-contributor-insights \
```



```
--max-results 4
```

Output:

```
{
  "ContributorInsightsSummaries": [
    {
      "TableName": "MusicCollection",
      "IndexName": "AlbumTitle-index",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "ProductCatalog",
      "ContributorInsightsStatus": "ENABLED"
    },
    {
      "TableName": "Forum",
      "ContributorInsightsStatus": "ENABLED"
    }
  ],
  "NextToken":
  "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFghI2Jk3LmnoPQ6RST9"
}
```

Per ulteriori informazioni, consulta [Analyzing Data Access Using CloudWatch Contributor Insights for DynamoDB nella Amazon DynamoDB Developer Guide](#).

Esempio 3: per recuperare la pagina successiva dei risultati

Il comando seguente utilizza il NextToken valore di una precedente chiamata al list-contributor-insights comando per recuperare un'altra pagina di risultati. Poiché la risposta in questo caso non include un NextToken valore, sappiamo di aver raggiunto la fine dei risultati.

```
aws dynamodb list-contributor-insights \
  --max-results 4 \
  --next-
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFghI2Jk3LmnoPQ6RST9
```

Output:

```
{
```

```
"ContributorInsightsSummaries": [  
  {  
    "TableName": "Reply",  
    "ContributorInsightsStatus": "ENABLED"  
  },  
  {  
    "TableName": "Thread",  
    "ContributorInsightsStatus": "ENABLED"  
  }  
]
```

Per ulteriori informazioni, consulta [Analyzing Data Access Using CloudWatch Contributor Insights for DynamoDB nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [ListContributorInsights](#) AWS CLI

## list-global-tables

Il seguente esempio di codice mostra come utilizzare `list-global-tables`.

AWS CLI

Per elencare le tabelle globali DynamoDB esistenti

L'`list-global-tables` esempio seguente elenca tutte le tabelle globali esistenti.

```
aws dynamodb list-global-tables
```

Output:

```
{  
  "GlobalTables": [  
    {  
      "GlobalTableName": "MusicCollection",  
      "ReplicationGroup": [  
        {  
          "RegionName": "us-east-2"  
        },  
        {  
          "RegionName": "us-east-1"  
        }  
      ]  
    }  
  ]  
}
```

```
    ]
  }
]
}
```

Per ulteriori informazioni, consulta [DynamoDB Global Tables nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta [ListGlobalTables](#) Command Reference.AWS CLI

## list-tables

Il seguente esempio di codice mostra come utilizzare `list-tables`.

### AWS CLI

Esempio 1: Per elencare le tabelle

L'`list-tables` esempio seguente elenca tutte le tabelle associate all' AWS account corrente e alla regione.

```
aws dynamodb list-tables
```

Output:

```
{
  "TableNames": [
    "Forum",
    "ProductCatalog",
    "Reply",
    "Thread"
  ]
}
```

Per ulteriori informazioni, consulta [Listing Table Names](#) nella Amazon DynamoDB Developer Guide.

Esempio 2: limitare le dimensioni della pagina

L'esempio seguente restituisce un elenco di tutte le tabelle esistenti, ma recupera solo un elemento in ogni chiamata, eseguendo più chiamate se necessario per ottenere l'intero elenco.

La limitazione delle dimensioni della pagina è utile quando si eseguono comandi di elenco su un numero elevato di risorse, il che può causare un errore di «timeout» quando si utilizza la dimensione di pagina predefinita di 1000.

```
aws dynamodb list-tables \  
  --page-size 1
```

Output:

```
{  
  "TableNames": [  
    "Forum",  
    "ProductCatalog",  
    "Reply",  
    "Thread"  
  ]  
}
```

Per ulteriori informazioni, consulta [Listing Table Names](#) nella Amazon DynamoDB Developer Guide.

Esempio 3: limitare il numero di articoli restituiti

L'esempio seguente limita il numero di articoli restituiti a 2. La risposta include un NextToken valore con cui recuperare la pagina successiva di risultati.

```
aws dynamodb list-tables \  
  --max-items 2
```

Output:

```
{  
  "TableNames": [  
    "Forum",  
    "ProductCatalog"  
  ],  
  "NextToken":  
  "abCDeFGhIJKlMnOPqrSTuvwXYZ1aBCdEFghijK7LM51n0pqRSTuv3WxY3ZabC5dEFghI2Jk3LmnoPQ6RST9"  
}
```

Per ulteriori informazioni, consulta [Listing Table Names](#) nella Amazon DynamoDB Developer Guide.

Esempio 4: per recuperare la pagina successiva dei risultati

Il comando seguente utilizza il NextToken valore di una precedente chiamata al `list-tables` comando per recuperare un'altra pagina di risultati. Poiché la risposta in questo caso non include un NextToken valore, sappiamo di aver raggiunto la fine dei risultati.

```
aws dynamodb list-tables \
  --starting-
  token abCDeFGhiJKLmnOPqrSTuvwxYZ1aBCdEFghIjK7LM51n0pqRSTuv3WxY3ZabC5dEFghI2Jk3LmnoPQ6RST9
```

Output:

```
{
  "TableNames": [
    "Reply",
    "Thread"
  ]
}
```

Per ulteriori informazioni, consulta [Listing Table Names](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListTables](#)Reference.

## list-tags-of-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-of-resource`.

AWS CLI

Esempio 1: elencare i tag di una risorsa DynamoDB

L'`list-tags-of-resource` esempio seguente visualizza i tag per la `MusicCollection` tabella.

```
aws dynamodb list-tags-of-resource \
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection
```

**Output:**

```
{
  "Tags": [
    {
      "Key": "Owner",
      "Value": "blueTeam"
    },
    {
      "Key": "Environment",
      "Value": "Production"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Tagging for DynamoDB nella Amazon DynamoDB Developer Guide](#).

**Esempio 2: limitare il numero di tag restituiti**

L'esempio seguente limita il numero di tag restituiti a 1. La risposta include un NextToken valore con cui recuperare la pagina successiva di risultati.

```
aws dynamodb list-tags-of-resource \
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
  --max-items 1
```

**Output:**

```
{
  "Tags": [
    {
      "Key": "Owner",
      "Value": "blueTeam"
    }
  ],
  "NextToken":
  "abCDeFGhiJKlMnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

Per ulteriori informazioni, consulta [Tagging for DynamoDB nella Amazon DynamoDB Developer Guide](#).

### Esempio 3: per recuperare la pagina successiva dei risultati

Il comando seguente utilizza il NextToken valore di una precedente chiamata al `list-tags-of-resource` comando per recuperare un'altra pagina di risultati. Poiché la risposta in questo caso non include un NextToken valore, sappiamo di aver raggiunto la fine dei risultati.

```
aws dynamodb list-tags-of-resource \  
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \  
  --starting-  
token abCDeFGhiJKLmnOPqrSTuvwXYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

Output:

```
{  
  "Tags": [  
    {  
      "Key": "Environment",  
      "Value": "Production"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Tagging for DynamoDB nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [ListTagsOfResource](#) AWS CLI

## put-item

Il seguente esempio di codice mostra come utilizzare `put-item`.

AWS CLI

Esempio 1: aggiungere un elemento a una tabella

L'`put-item` esempio seguente aggiunge un nuovo elemento alla `MusicCollection` tabella.

```
aws dynamodb put-item \  
  --table-name MusicCollection \  
  --item file://item.json \  
  --return-consumed-capacity TOTAL \  

```

```
--return-item-collection-metrics SIZE
```

Contenuto di `item.json`.

```
{
  "Artist": {"S": "No One You Know"},
  "SongTitle": {"S": "Call Me Today"},
  "AlbumTitle": {"S": "Greatest Hits"}
}
```

Output:

```
{
  "ConsumedCapacity": {
    "TableName": "MusicCollection",
    "CapacityUnits": 1.0
  },
  "ItemCollectionMetrics": {
    "ItemCollectionKey": {
      "Artist": {
        "S": "No One You Know"
      }
    },
    "SizeEstimateRangeGB": [
      0.0,
      1.0
    ]
  }
}
```

Per ulteriori informazioni, consulta [Writing an Item](#) in Amazon DynamoDB Developer Guide.

Esempio 2: sovrascrivere in modo condizionale un elemento in una tabella

L'`put-item` seguente sovrascrive un elemento esistente nella `MusicCollection` tabella solo se tale elemento esistente ha un `AlbumTitle` attributo con un valore di `Greatest Hits`. Il comando restituisce il valore precedente dell'elemento.

```
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
```

Contenuto di `item.json`.

```
{  
  "Artist": {"S": "No One You Know"},  
  "SongTitle": {"S": "Call Me Today"},  
  "AlbumTitle": {"S": "Somewhat Famous"}  
}
```

Contenuto di `names.json`.

```
{  
  "#A": "AlbumTitle"  
}
```

Contenuto di `values.json`.

```
{  
  ":A": {"S": "Greatest Hits"}  
}
```

Output:

```
{  
  "Attributes": {  
    "AlbumTitle": {  
      "S": "Greatest Hits"  
    },  
    "Artist": {  
      "S": "No One You Know"  
    },  
    "SongTitle": {  
      "S": "Call Me Today"  
    }  
  }  
}
```

Se la chiave esiste già, dovresti vedere il seguente risultato:

A client error (ConditionalCheckFailedException) occurred when calling the PutItem operation: The conditional request failed.

Per ulteriori informazioni, consulta [Writing an Item](#) in Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [PutItem](#)Reference.

## query

Il seguente esempio di codice mostra come utilizzare query.

### AWS CLI

Esempio 1: interrogare una tabella

L'queryesempio seguente interroga gli elementi della MusicCollection tabella. La tabella ha una chiave hash-and-range primaria (ArtisteSongTitle), ma questa query specifica solo il valore della chiave hash. Restituisce i titoli delle canzoni dell'artista chiamato «No One You Know».

```
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
```

Contenuto di expression-attributes.json.

```
{  
  ":v1": {"S": "No One You Know"}  
}
```

Output:

```
{  
  "Items": [  
    {  
      "SongTitle": {  
        "S": "Call Me Today"      }  
    }  
  ]  
}
```

```

    },
    "SongTitle": {
      "S": "Scared of My Shadow"
    }
  }
],
"Count": 2,
"ScannedCount": 2,
"ConsumedCapacity": {
  "TableName": "MusicCollection",
  "CapacityUnits": 0.5
}
}

```

Per ulteriori informazioni, consulta [Working with Queries in DynamoDB nella Amazon DynamoDB Developer Guide](#).

Esempio 2: interrogare una tabella utilizzando letture fortemente coerenti e attraversare l'indice in ordine decrescente

L'esempio seguente esegue la stessa query del primo esempio, ma restituisce i risultati in ordine inverso e utilizza letture fortemente coerenti.

```

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

```

Contenuto di `expression-attributes.json`.

```

{
  ":v1": {"S": "No One You Know"}
}

```

Output:

```

{

```

```

    "Items": [
      {
        "SongTitle": {
          "S": "Scared of My Shadow"
        }
      },
      {
        "SongTitle": {
          "S": "Call Me Today"
        }
      }
    ],
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": {
      "TableName": "MusicCollection",
      "CapacityUnits": 1.0
    }
  }
}

```

Per ulteriori informazioni, consulta [Working with Queries in DynamoDB nella Amazon DynamoDB Developer Guide](#).

Esempio 3: per filtrare risultati specifici

L'esempio seguente interroga MusicCollection ma esclude i risultati con valori specifici nell'AlbumTitle attributo. Si noti che ciò non influisce sull'ScannedCount o ConsumedCapacity, poiché il filtro viene applicato dopo la lettura degli elementi.

```

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

```

Contenuto di values.json.

```

{
  ":v1": {"S": "No One You Know"},

```

```

    ":v2": {"S": "Blue Sky Blues"},
    ":v3": {"S": "Greatest Hits"}
  }

```

Contenuto di `names.json`.

```

{
  "#n1": "Artist",
  "#n2": "AlbumTitle"
}

```

Output:

```

{
  "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
  }
}

```

Per ulteriori informazioni, consulta [Working with Queries in DynamoDB nella Amazon DynamoDB Developer Guide](#).

Esempio 4: per recuperare solo il numero di articoli

L'esempio seguente recupera il conteggio degli elementi corrispondenti alla query, ma non recupera nessuno degli elementi stessi.

```
aws dynamodb query \  
  --table-name MusicCollection \  
  --select COUNT \  
  --key-condition-expression "Artist = :v1" \  
  --expression-attribute-values file://expression-attributes.json
```

Contenuto di `expression-attributes.json`.

```
{  
  ":v1": {"S": "No One You Know"}  
}
```

Output:

```
{  
  "Count": 2,  
  "ScannedCount": 2,  
  "ConsumedCapacity": null  
}
```

Per ulteriori informazioni, consulta [Working with Queries in DynamoDB nella Amazon DynamoDB Developer Guide](#).

Esempio 5: interrogare un indice

L'esempio seguente esegue una query sull'indice `AlbumTitleIndex` secondario locale. La query restituisce tutti gli attributi della tabella di base che sono stati proiettati nell'indice secondario locale. Si noti che quando si esegue una query su un indice secondario locale o su un indice secondario globale, è necessario fornire anche il nome della tabella di base utilizzando il `table-name` parametro.

```
aws dynamodb query \  
  --table-name MusicCollection \  
  --index-name AlbumTitleIndex \  
  --key-condition-expression "Artist = :v1" \  
  --expression-attribute-values file://expression-attributes.json \  
  --select ALL_PROJECTED_ATTRIBUTES \  
  --return-consumed-capacity INDEXES
```

Contenuto di `expression-attributes.json`.

```
{
  ":v1": {"S": "No One You Know"}
}
```

**Output:**

```
{
  "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
    }
  }
}
```

```
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Working with Queries in DynamoDB nella Amazon DynamoDB Developer Guide](#).

- [Per i API dettagli, consulta Query in Command Reference.AWS CLI](#)

## restore-table-from-backup

Il seguente esempio di codice mostra come utilizzare `restore-table-from-backup`.

### AWS CLI

Per ripristinare una tabella DynamoDB da un backup esistente

L'esempio seguente ripristina la tabella specificata da un backup esistente.

```
aws dynamodb restore-table-from-backup \  
  --target-table-name MusicCollection \  
  --backup-arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/  
  backup/01576616366715-b4e58d3a
```

Output:

```
{  
  "TableDescription": {  
    "AttributeDefinitions": [  
      {  
        "AttributeName": "Artist",  
        "AttributeType": "S"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "AttributeType": "S"  
      }  
    ],  
    "TableName": "MusicCollection2",  
    "KeySchema": [  
      {  
        "AttributeName": "Artist",  
        "KeyType": "HASH"  
      },  
      {  
        "AttributeName": "SongTitle",  
        "KeyType": "RANGE"  
      }  
    ]  
  }  
}
```



```

    {
      "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
  }
}

```

Per ulteriori informazioni, consulta [Backup e ripristino su richiesta per DynamoDB nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [RestoreTableFromBackup](#)AWS CLI

## restore-table-to-point-in-time

Il seguente esempio di codice mostra come utilizzare `restore-table-to-point-in-time`.

## AWS CLI

Per ripristinare una tabella DynamoDB in un determinato momento

L'`restore-table-to-point-in-time` seguente ripristina la `MusicCollection` tabella nel momento specificato.

```
aws dynamodb restore-table-to-point-in-time \  
  --source-table-name MusicCollection \  
  --target-table-name MusicCollectionRestore \  
  --restore-date-time 1576622404.0
```

Output:

```
{  
  "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
    }
  }
}
}

```

Per ulteriori informazioni, consulta [Point-in-Time Recovery for DynamoDB](#) nella Amazon DynamoDB Developer Guide.

- Per i API dettagli, consulta [RestoreTableToPointInTime](#) Command Reference.AWS CLI

## scan

Il seguente esempio di codice mostra come utilizzare `scan`.

### AWS CLI

Per scansionare una tabella

L'esempio seguente esegue la scansione dell'intera `MusicCollection` tabella, quindi restringe i risultati alle canzoni dell'artista «No One You Know». Per ogni elemento, vengono restituiti solo il titolo dell'album e il titolo del brano.

```

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

```

Contenuto di `expression-attribute-names.json`.

```
{
  "#ST": "SongTitle",
  "#AT": "AlbumTitle"
}
```

Contenuto di `expression-attribute-values.json`.

```
{
  ":a": {"S": "No One You Know"}
}
```

Output:

```
{
  "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
}
```

Per ulteriori informazioni, consulta [Working with Scans in DynamoDB nella Amazon DynamoDB Developer Guide](#).

- [Per i API dettagli, consulta Scan in Command Reference.AWS CLI](#)

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere tag a una risorsa DynamoDB

L'esempio seguente aggiunge una coppia chiave/valore di tag alla tabella `MusicCollection`

```
aws dynamodb tag-resource \  
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \  
  --tags Key=Owner,Value=blueTeam
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging for DynamoDB nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [TagResource](#) AWS CLI

## transact-get-items

Il seguente esempio di codice mostra come utilizzare `transact-get-items`.

### AWS CLI

Per recuperare più elementi atomicamente da una o più tabelle

L'esempio seguente recupera più elementi in modo atomico.

```
aws dynamodb transact-get-items \  
  --transact-items file://transact-items.json \  
  --return-consumed-capacity TOTAL
```

Contenuto di `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"
    }
}
]

```

### Output:

```

{
  "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"
        }
    }
}
]
}

```

Per ulteriori informazioni, consulta [Managing Complex Workflows with DynamoDB Transactions nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [TransactGetItems](#) AWS CLI

## transact-write-items

Il seguente esempio di codice mostra come utilizzare `transact-write-items`.

### AWS CLI

Esempio 1: scrivere elementi atomicamente su una o più tabelle

L'`transact-write-items` esempio seguente aggiorna un elemento e ne elimina un altro. L'operazione ha esito negativo se una delle operazioni fallisce o se uno degli elementi contiene un `Rating` attributo.

```

aws dynamodb transact-write-items \
  --transact-items file://transact-items.json \
  --return-consumed-capacity TOTAL \
  --return-item-collection-metrics SIZE

```

Contenuto del `transact-items.json` file:

```

[
  {
    "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)"
    }
}
]

```

### Output:

```

{
    "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
      ]
    }
  ]
}

```

Per ulteriori informazioni, consulta [Managing Complex Workflows with DynamoDB Transactions nella Amazon DynamoDB Developer Guide](#).

Esempio 2: scrivere elementi in modo atomico utilizzando un token di richiesta del cliente

Il comando seguente utilizza un token di richiesta client per effettuare la chiamata a `transact-write-items idempotent`, il che significa che più chiamate hanno lo stesso effetto di una singola chiamata.

```

aws dynamodb transact-write-items \
  --transact-items file://transact-items.json \
  --client-request-token abc123

```

Contenuto del file: `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)"
  }
}
]

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing Complex Workflows with DynamoDB Transactions nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [TransactWriteItems](#) AWS CLI

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da una risorsa DynamoDB

L'esempio seguente rimuove il tag con la chiave `Owner` dalla `MusicCollection` tabella.

```

aws dynamodb untag-resource \
  --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
  --tag-keys Owner

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging for DynamoDB nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [UntagResource](#)AWS CLI

## update-continuous-backups

Il seguente esempio di codice mostra come utilizzare `update-continuous-backups`.

### AWS CLI

Per aggiornare le impostazioni di backup continuo per una tabella DynamoDB

L'`update-continuous-backups`esempio seguente abilita point-in-time il ripristino della `MusicCollection` tabella.

```
aws dynamodb update-continuous-backups \  
  --table-name MusicCollection \  
  --point-in-time-recovery-specification PointInTimeRecoveryEnabled=true
```

Output:

```
{  
  "ContinuousBackupsDescription": {  
    "ContinuousBackupsStatus": "ENABLED",  
    "PointInTimeRecoveryDescription": {  
      "PointInTimeRecoveryStatus": "ENABLED",  
      "EarliestRestorableDateTime": 1576622404.0,  
      "LatestRestorableDateTime": 1576622404.0  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Point-in-Time Recovery for DynamoDB](#) nella Amazon DynamoDB Developer Guide.

- Per i API dettagli, consulta [UpdateContinuousBackups](#)Command Reference.AWS CLI

## update-contributor-insights

Il seguente esempio di codice mostra come utilizzare `update-contributor-insights`.

### AWS CLI

Per abilitare Contributor Insights su una tabella

L'update-contributor-insightsempio seguente abilita Contributor Insights nella MusicCollection tabella e nell'indice secondario AlbumTitle-index globale.

```
aws dynamodb update-contributor-insights \  
  --table-name MusicCollection \  
  --index-name AlbumTitle-index \  
  --contributor-insights-action ENABLE
```

Output:

```
{  
  "TableName": "MusicCollection",  
  "IndexName": "AlbumTitle-index",  
  "ContributorInsightsStatus": "ENABLING"  
}
```

Per ulteriori informazioni, consulta [Analyzing Data Access Using CloudWatch Contributor Insights for DynamoDB nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [UpdateContributorInsights](#) AWS CLI

## update-global-table-settings

Il seguente esempio di codice mostra come utilizzare update-global-table-settings.

AWS CLI

Per aggiornare le impostazioni della capacità di scrittura assegnata su una tabella globale DynamoDB

L'update-global-table-settingsempio seguente imposta la capacità di scrittura assegnata alla tabella MusicCollection globale su 15.

```
aws dynamodb update-global-table-settings \  
  --global-table-name MusicCollection \  
  --global-table-provisioned-write-capacity-units 15
```

Output:

```
{  
  "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  
    }  
  }  
]  
}
```

Per ulteriori informazioni, consulta [DynamoDB Global Tables nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta [UpdateGlobalTableSettings](#) Command Reference.AWS CLI

## update-global-table

Il seguente esempio di codice mostra come utilizzare `update-global-table`.

### AWS CLI

Per aggiornare una tabella globale DynamoDB

L'esempio seguente aggiunge una replica nella regione specificata alla `MusicCollection` tabella globale.

```
aws dynamodb update-global-table \  
  --global-table-name MusicCollection \  
  --replica-updates Create={RegionName=eu-west-1}
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [DynamoDB Global Tables nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta [UpdateGlobalTable](#) Command Reference. AWS CLI

## update-item

Il seguente esempio di codice mostra come utilizzare `update-item`.

### AWS CLI

Esempio 1: aggiornare un elemento in una tabella

L'esempio `update-item` seguente legge una voce dalla tabella `MusicCollection`. Aggiunge un nuovo attributo (`Year`) e modifica l'`AlbumTitle` attributo. Tutti gli attributi dell'elemento, così come appaiono dopo l'aggiornamento, vengono restituiti nella risposta.

```
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
```

Contenuto di `key.json`.

```
{  
  "Artist": {"S": "Acme Band"},  
  "SongTitle": {"S": "Happy Day"}  
}
```

Contenuto di `expression-attribute-names.json`.

```
{  
  "#Y": "Year", "#AT": "AlbumTitle"  
}
```

Contenuto di `expression-attribute-values.json`.

```
{  
  ":y": {"N": "2015"},  
  ":t": {"S": "Louder Than Ever"}  
}
```

**Output:**

```
{
  "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
  ]
}
```

Per ulteriori informazioni, consulta [Writing an Item](#) in Amazon DynamoDB Developer Guide.

**Esempio 2: aggiornare un articolo in modo condizionale**

L'esempio seguente aggiorna un elemento nella MusicCollection tabella, ma solo se l'elemento esistente non ha già un Year attributo.



```
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)"
```

Contenuto di `key.json`.

```
{  
  "Artist": {"S": "Acme Band"},  
  "SongTitle": {"S": "Happy Day"}  
}
```

Contenuto di `expression-attribute-names.json`.

```
{  
  "#Y": "Year",  
  "#AT": "AlbumTitle"  
}
```

Contenuto di `expression-attribute-values.json`.

```
{  
  ":y": {"N": "2015"},  
  ":t": {"S": "Louder Than Ever"}  
}
```

Se l'elemento ha già un `Year` attributo, DynamoDB restituisce il seguente output.

```
An error occurred (ConditionalCheckFailedException) when calling the UpdateItem  
operation: The conditional request failed
```

Per ulteriori informazioni, consulta [Writing an Item](#) in Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateItem](#) Reference.

## update-table-replica-auto-scaling

Il seguente esempio di codice mostra come utilizzare `update-table-replica-auto-scaling`.

## AWS CLI

Per aggiornare le impostazioni di ridimensionamento automatico tra le repliche di una tabella globale

L'update-table-replica-auto-scaling esempio seguente aggiorna le impostazioni di auto scaling della capacità di scrittura tra le repliche della tabella globale specificata.

```
aws dynamodb update-table-replica-auto-scaling \  
  --table-name MusicCollection \  
  --provisioned-write-capacity-auto-scaling-update file://auto-scaling-policy.json
```

Contenuto di auto-scaling-policy.json.

```
{  
  "MinimumUnits": 10,  
  "MaximumUnits": 100,  
  "AutoScalingDisabled": false,  
  "ScalingPolicyUpdate": {  
    "PolicyName": "DynamoDBWriteCapacityUtilization:table/MusicCollection",  
    "TargetTrackingScalingPolicyConfiguration": {  
      "TargetValue": 80  
    }  
  }  
}
```

Output:

```
{  
  "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"
    }
]
}
}
}

```

Per ulteriori informazioni, consulta [DynamoDB Global Tables nella Amazon DynamoDB Developer Guide](#).

- Per i API dettagli, consulta [UpdateTableReplicaAutoScaling](#) Command Reference.AWS CLI

## update-table

Il seguente esempio di codice mostra come utilizzare `update-table`.

### AWS CLI

Esempio 1: modificare la modalità di fatturazione di una tabella

L'update-table esempio seguente aumenta la capacità di lettura e scrittura assegnata alla `MusicCollection` tabella.

```

aws dynamodb update-table \
  --table-name MusicCollection \
  --billing-mode PROVISIONED \
  --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10

```

Output:

```
{
```

```
"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"
  }
}
```

```
}
```

Per ulteriori informazioni, consulta [Updating a Table](#) nella Amazon DynamoDB Developer Guide.

Esempio 2: creare un indice secondario globale

L'esempio seguente aggiunge un indice secondario globale alla `MusicCollection` tabella.

```
aws dynamodb update-table \  
  --table-name MusicCollection \  
  --attribute-definitions AttributeName=AlbumTitle,AttributeType=S \  
  --global-secondary-index-updates file://gsi-updates.json
```

Contenuto di `gsi-updates.json`.

```
[  
  {  
    "Create": {  
      "IndexName": "AlbumTitle-index",  
      "KeySchema": [  
        {  
          "AttributeName": "AlbumTitle",  
          "KeyType": "HASH"  
        }  
      ],  
      "ProvisionedThroughput": {  
        "ReadCapacityUnits": 10,  
        "WriteCapacityUnits": 10  
      },  
      "Projection": {  
        "ProjectionType": "ALL"  
      }  
    }  
  }  
]
```

Output:

```
{  
  "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"
  }
]
}
}

```

Per ulteriori informazioni, consulta [Updating a Table](#) nella Amazon DynamoDB Developer Guide.

### Esempio 3: abilitare DynamoDB Streams su una tabella

Il comando seguente abilita DynamoDB Streams sulla tabella. `MusicCollection`

```

aws dynamodb update-table \
  --table-name MusicCollection \
  --stream-specification StreamEnabled=true,StreamViewType=NEW_IMAGE

```

Output:

```

{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Updating a Table](#) nella Amazon DynamoDB Developer Guide.

Esempio 4: Per abilitare la crittografia lato server

L'esempio seguente abilita la crittografia lato server sulla tabella. MusicCollection

```
aws dynamodb update-table \  
  --table-name MusicCollection \  
  --sse-specification Enabled=true,SSEType=KMS
```

Output:

```
{  
  "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"
    }
  }
}

```

Per ulteriori informazioni, consulta [Updating a Table](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateTableReference](#).

## update-time-to-live

Il seguente esempio di codice mostra come utilizzare `update-time-to-live`.

### AWS CLI

Per aggiornare le impostazioni Time to Live su una tabella

L'`update-time-to-live` esempio seguente abilita Time to Live nella tabella specificata.

```
aws dynamodb update-time-to-live \  
  --table-name MusicCollection \  
  --time-to-live-specification Enabled=true,AttributeName=t1
```

Output:

```
{  
  "TimeToLiveSpecification": {  
    "Enabled": true,  
    "AttributeName": "t1"  
  }  
}
```

Per ulteriori informazioni, consulta [Time to Live](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateTimeToLive](#) Reference.

## Esempi di DynamoDB Streams che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface con DynamoDB Streams.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **describe-stream**

Il seguente esempio di codice mostra come utilizzare `describe-stream`.

#### AWS CLI

Per ottenere informazioni su uno stream DynamoDB

Il `describe-stream` comando seguente visualizza informazioni sullo specifico flusso DynamoDB.

```
aws dynamodbstreams describe-stream \  
  --stream-arn arn:aws:dynamodb:us-west-1:123456789012:table/Music/  
stream/2019-10-22T18:02:01.576
```

Output:

```
{  
  "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"
  }
]
}
}

```

Per ulteriori informazioni, consulta [Capturing Table Activity with DynamoDB Streams](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command Reference. [DescribeStream](#) AWS CLI

## get-records

Il seguente esempio di codice mostra come utilizzare `get-records`.

### AWS CLI

Per ottenere record da uno stream Dynamodb

Il `get-records` comando seguente recupera i record utilizzando l'iteratore di shard Amazon DynamoDB specificato.

```

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
+hFxFAWR5C7QILOXPc8mRBfNIazfrVCjJK8/jsjCzsQnyXKzJbhh+GXCoxYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaXNfThDwRAYvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPIdmTRG
+w/LEGS05ha1qNP+VL4+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5KPyw+DtcM5BHayBs
+PVYHITaTLiInFLT
+LCwvaz1QH3MY3b8A05Z800wjpkM60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHnndusw==

```

## Output:

```
{
  "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|AAAAAAAAAAAAEhEI04jkFLW
+LK0wivjT8d/IHEh3iExV2xK00aTxEzVy1C1C7Kbb5+Z0W6bT9VQ2n1/
mrs7+PRia0ZCHJu7JHJVW7zlsq0i/ges3fw8GYEymyL+piEk35cx67rQqwKKyq
+Q6w9JyjreI0j4F2lWLv261BwRTriYC4IB7C3BZZK4715QwYdXNdVHiSBRZX8UqoS6W0t0F87xZLNB9F/
NhYBLXi/wcGvAcBcC0TNI0H+N0Nqwt0B/
FGckNrf8YZ0xRoNN6RgGuVWHF3px0hxEJeFZoSoJTIKeG9YcYxzi5Ci/
mhdTm7tBXnbw5c6xmsGsBqTirNjldyJLcWl8Cl0U0LX63Ufo/5QliztcjEbKsQe28x8LM8o7VH1Is0FF/

```

```
ITt8awSA4igyJS0P87GN8Qri8kj8iaE35805jBHWf2wvwT6Iy2xGrR2r2HzYps9dwG0arVdEITaJfWzNoL4HajMhmREZ
+V04i1YIeHMXJfcwetNRuIbdQXfJht2NQZa4PVV6iknY6d19MrdbSTMKoqAuvp6g3Q2jH4t7GKCLWgodcPAn8g5+43Da
}
```

Per ulteriori informazioni, consulta [Capturing Table Activity with DynamoDB Streams](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command Reference. [GetRecords](#) AWS CLI

## get-shard-iterator

Il seguente esempio di codice mostra come utilizzare `get-shard-iterator`.

### AWS CLI

Per ottenere uno shard iterator

Il `get-shard-iterator` comando seguente recupera un iteratore di shard per lo shard specificato.

```
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-0000001571780995058-40810d86 \
  --shard-iterator-type LATEST
```

Output:

```
{
  "ShardIterator": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576|1|
AAAAAAAAAAGgM3YZ89vLZZxjmoQeo33r9M4x3+zmmTLsiL86MfrF4+B4EbsByi52InVmi0Nmy6xVW4IRcIIbs1z07MNI
+CjNP1qQjnyRSAnf0wWmKhL1/KNParWSfz2odf780o00bIDIWRRMkt7+Hyzh9SD
+hFxFAWR5C7QI10XPc8mRBfNIazfrVCjJK8/jsjCzsqNyXKzJbh+hGXCoXYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaXNfThDwRAYvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPI dmTRG
+w/1EGS05ha1qNP+V14+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5KPwy+Dtcm5BHAYBs
+PVYHITaTliInFlT
+LCwvaz1QH3MY3b8A05Z800wjpktm60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHnndusw==
}
```

Per ulteriori informazioni, consulta [Capturing Table Activity with DynamoDB Streams](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command Reference. [GetShardIterator](#)AWS CLI

## list-streams

Il seguente esempio di codice mostra come utilizzare `list-streams`.

### AWS CLI

Per elencare i flussi DynamoDB

Il `list-streams` comando seguente elenca tutti i flussi Amazon DynamoDB esistenti all'interno della regione predefinita. AWS

```
aws dynamodbstreams list-streams
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Capturing Table Activity with DynamoDB Streams](#) nella Amazon DynamoDB Developer Guide.

- Per API i dettagli, consulta Command Reference. [ListStreams](#)AWS CLI

## EC2Esempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con AmazonEC2.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **accept-address-transfer**

Il seguente esempio di codice mostra come utilizzare `accept-address-transfer`.

#### AWS CLI

Accettare un indirizzo IP elastico trasferito sul tuo account

L'esempio seguente accetta il trasferimento dell'indirizzo IP elastico specificato al tuo account.

```
aws ec2 accept-address-transfer \
  --address 100.21.184.216
```

#### Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Transfer Elastic IP address](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [AcceptAddressTransfer AWS CLI Command Reference](#).

## accept-reserved-instances-exchange-quote

Il seguente esempio di codice mostra come utilizzare `accept-reserved-instances-exchange-quote`.

### AWS CLI

Per eseguire uno scambio di istanze riservate convertibili

Questo esempio esegue uno scambio delle istanze riservate convertibili specificate.

Comando:

```
aws ec2 accept-reserved-instances-exchange-quote --reserved-  
instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-  
configurations OfferingId=b747b472-423c-48f3-8cee-679bcexample
```

Output:

```
{  
  "ExchangeId": "riex-e68ed3c1-8bc8-4c17-af77-811afexample"  
}
```

- Per API i dettagli, vedere [AcceptReservedInstancesExchangeQuote](#) in AWS CLI Command Reference.

## accept-transit-gateway-peering-attachment

Il seguente esempio di codice mostra come utilizzare `accept-transit-gateway-peering-attachment`.

### AWS CLI

Accettare un allegato di peering del gateway di transito

L'esempio seguente accetta l'allegato di peering del gateway di transito specificato. Il `--region` parametro specifica la regione in cui si trova il gateway di transito accettante.

```
aws ec2 accept-transit-gateway-peering-attachment \  
  --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \  
  --region us-east-1
```



```
--region us-east-2
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Transit Gateway Peering Attachments](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [AcceptTransitGatewayPeeringAttachment](#) in AWS CLI Command Reference.

## accept-transit-gateway-vpc-attachment

Il seguente esempio di codice mostra come utilizzare `accept-transit-gateway-vpc-attachment`.

### AWS CLI

Accettare una richiesta di collegamento a VPC a un gateway di transito.

L'esempio seguente accetta la richiesta come allegato specificato.

```
aws ec2 accept-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

**Output:**

```
{
  "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"
    }
  }
}
```

Per ulteriori informazioni, consulta [Transit Gateway Attachments to a VPC](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [AcceptTransitGatewayVpcAttachment](#) in AWS CLI Command Reference.

**accept-vpc-endpoint-connections**

Il seguente esempio di codice mostra come utilizzare `accept-vpc-endpoint-connections`.

**AWS CLI**

Accettare una richiesta di connessione all'endpoint dell'interfaccia

Questo esempio accetta la richiesta di connessione all'endpoint specificata per il servizio endpoint specificato.

Comando:

```
aws ec2 accept-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --
vpc-endpoint-ids vpce-0c1308d7312217abc
```

**Output:**

```
{
  "Unsuccessful": []
}
```

- Per API i dettagli, vedere [AcceptVpcEndpointConnections](#) in AWS CLI Command Reference.

**accept-vpc-peering-connection**

Il seguente esempio di codice mostra come utilizzare `accept-vpc-peering-connection`.

**AWS CLI**

Accettare una connessione VPC peering

Questo esempio accetta la richiesta di connessione VPC peering specificata.

Comando:

```
aws ec2 accept-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

**Output:**

```
{
  "VpcPeeringConnection": {
    "Status": {
      "Message": "Provisioning",
      "Code": "provisioning"
    },
    "Tags": [],
    "AcceptorVpcInfo": {
      "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"
    }
  }
}
```

```
}  
}
```

- Per API i dettagli, vedere [AcceptVpcPeeringConnection](#) in AWS CLI Command Reference.

## advertise-byoip-cidr

Il seguente esempio di codice mostra come utilizzare `advertise-byoip-cidr`.

### AWS CLI

Per pubblicizzare un intervallo di indirizzi

L'esempio seguente pubblica l'intervallo di IPv4 indirizzi pubblici specificato.

```
aws ec2 advertise-byoip-cidr \  
  --cidr 203.0.113.25/24
```

Output:

```
{  
  "ByoipCidr": {  
    "Cidr": "203.0.113.25/24",  
    "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",  
    "State": "provisioned"  
  }  
}
```

- Per API i dettagli, vedere [AdvertiseByoipCidr](#) in AWS CLI Command Reference.

## allocate-address

Il seguente esempio di codice mostra come utilizzare `allocate-address`.

### AWS CLI

Esempio 1: per allocare un indirizzo IP elastico dal pool di indirizzi Amazon

Nell'esempio di `allocate-address` seguente viene allocato un indirizzo IP elastico. Amazon EC2 seleziona l'indirizzo dal pool di indirizzi di Amazon.

```
aws ec2 allocate-address
```

Output:

```
{
  "PublicIp": "70.224.234.241",
  "AllocationId": "eipalloc-01435ba59eEXAMPLE",
  "PublicIpv4Pool": "amazon",
  "NetworkBorderGroup": "us-west-2",
  "Domain": "vpc"
}
```

Per ulteriori informazioni, consulta [gli indirizzi IP elastici](#) nella Amazon EC2 User Guide.

Esempio 2: per allocare un indirizzo IP elastico e associarlo a un gruppo di confine di rete

Nell'esempio di `allocate-address` seguente viene allocato un indirizzo IP elastico e viene associato al gruppo di confine di rete specificato.

```
aws ec2 allocate-address \
  --network-border-group us-west-2-lax-1
```

Output:

```
{
  "PublicIp": "70.224.234.241",
  "AllocationId": "eipalloc-e03dd489ceEXAMPLE",
  "PublicIpv4Pool": "amazon",
  "NetworkBorderGroup": "us-west-2-lax-1",
  "Domain": "vpc"
}
```

Per ulteriori informazioni, consulta [gli indirizzi IP elastici](#) nella Amazon EC2 User Guide.

Esempio 3: per allocare un indirizzo IP elastico da un pool di indirizzi proprietario

Nell'esempio di `allocate-address` seguente viene allocato un indirizzo IP elastico da un pool di indirizzi trasferito sull'account Amazon Web Services. Amazon EC2 seleziona l'indirizzo dal pool di indirizzi.

```
aws ec2 allocate-address \  
  --public-ipv4-pool ipv4pool-ec2-1234567890abcdef0
```

Output:

```
{  
  "AllocationId": "eipalloc-02463d08ceEXAMPLE",  
  "NetworkBorderGroup": "us-west-2",  
  "CustomerOwnedIp": "18.218.95.81",  
  "CustomerOwnedIpv4Pool": "ipv4pool-ec2-1234567890abcdef0",  
  "Domain": "vpc"  
  "NetworkBorderGroup": "us-west-2",  
}
```

Per ulteriori informazioni, consulta [gli indirizzi IP elastici](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [AllocateAddress AWS CLI Command Reference](#).

## allocate-hosts

Il seguente esempio di codice mostra come utilizzare `allocate-hosts`.

### AWS CLI

Esempio 1: allocare un host dedicato

L'`allocate-hosts` seguente alloca un singolo host dedicato nella zona di `eu-west-1a` disponibilità, su cui è possibile avviare `m5.large` le istanze. Per impostazione predefinita, l'host dedicato accetta solo l'avvio di istanze di destinazione e non supporta il ripristino dell'host.

```
aws ec2 allocate-hosts \  
  --instance-type m5.large \  
  --availability-zone eu-west-1a \  
  --quantity 1
```

Output:

```
{  
  "HostIds": [  
    "h-07879acf49EXAMPLE"  
  ]  
}
```

```
}
```

Esempio 2: allocare un host dedicato con posizionamento automatico e ripristino dell'host abilitati

L'`allocate-hosts` seguente alloca un singolo host dedicato nella zona di `eu-west-1a` disponibilità con posizionamento automatico e ripristino dell'host abilitati.

```
aws ec2 allocate-hosts \  
  --instance-type m5.large \  
  --availability-zone eu-west-1a \  
  --auto-placement on \  
  --host-recovery on \  
  --quantity 1
```

Output:

```
{  
  "HostIds": [  
    "h-07879acf49EXAMPLE"  
  ]  
}
```

Esempio 3: allocare un host dedicato con tag

L'`allocate-hosts` seguente alloca un singolo host dedicato e applica un tag con una chiave denominata `purpose` e un valore di `production`

```
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}'
```

Output:

```
{  
  "HostIds": [  
    "h-07879acf49EXAMPLE"  
  ]  
}
```

Per ulteriori informazioni, consulta [Allocazione di host dedicati](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per API i dettagli, consulta Command [AllocateHostsReference](#) AWS CLI .

## **allocate-ipam-pool-cidr**

Il seguente esempio di codice mostra come utilizzare `allocate-ipam-pool-cidr`.

### AWS CLI

Allocare un CIDR da un pool IPAM

L'`allocate-ipam-pool-cidr` seguente alloca un CIDR da un IPAM pool.

(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
```

Output:

```
{  
  "IpamPoolAllocation": {  
    "Cidr": "10.0.0.0/24",  
    "IpamPoolAllocationId": "ipam-pool-alloc-018ecc28043b54ba38e2cd99943cebfb",  
    "ResourceType": "custom",  
    "ResourceOwner": "123456789012"  
  }  
}
```

Per ulteriori informazioni, consulta [Allocazione manuale di un CIDR a un pool per riservare lo spazio degli indirizzi IP](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [AllocateIpamPoolCidr](#) AWS CLI Command Reference.



## apply-security-groups-to-client-vpn-target-network

Il seguente esempio di codice mostra come utilizzare `apply-security-groups-to-client-vpn-target-network`.

### AWS CLI

Per applicare gruppi di sicurezza a una rete di destinazione per un VPN endpoint Client

L'esempio seguente applica il gruppo di sicurezza `sg-01f6e627a89f4db32` all'associazione tra la rete di destinazione specificata e l'VPN endpoint Client.

```
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
```

Output:

```
{
  "SecurityGroupIds": [
    "sg-01f6e627a89f4db32"
  ]
}
```

Per ulteriori informazioni, consulta [Target Networks](#) nella AWS Client VPN Administrator Guide.

- Per API i dettagli, vedere [ApplySecurityGroupsToClientVpnTargetNetwork](#) in AWS CLI Command Reference.

## assign-ipv6-addresses

Il seguente esempio di codice mostra come utilizzare `assign-ipv6-addresses`.

### AWS CLI

Per assegnare IPv6 indirizzi specifici a un'interfaccia di rete

Questo esempio assegna IPv6 gli indirizzi specificati all'interfaccia di rete specificata.

Comando:

```
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
```

Output:

```
{  
  "AssignedIpv6Addresses": [  
    "2001:db8:1234:1a00:3304:8879:34cf:4071",  
    "2001:db8:1234:1a00:9691:9503:25ad:1761"  
  ],  
  "NetworkInterfaceId": "eni-38664473"  
}
```

Per assegnare IPv6 gli indirizzi selezionati da Amazon a un'interfaccia di rete

Questo esempio assegna due IPv6 indirizzi all'interfaccia di rete specificata. Amazon assegna automaticamente questi IPv6 indirizzi dagli IPv6 indirizzi disponibili nell'intervallo di IPv6 CIDR blocchi della sottorete.

Comando:

```
aws ec2 assign-ipv6-addresses --network-interface-id eni-38664473 --ipv6-address-  
count 2
```

Output:

```
{  
  "AssignedIpv6Addresses": [  
    "2001:db8:1234:1a00:3304:8879:34cf:4071",  
    "2001:db8:1234:1a00:9691:9503:25ad:1761"  
  ],  
  "NetworkInterfaceId": "eni-38664473"  
}
```

- Per API i dettagli, consulta [AssignIpv6Addresses](#) in AWS CLI Command Reference.

## assign-private-ip-addresses

Il seguente esempio di codice mostra come utilizzare `assign-private-ip-addresses`.

## AWS CLI

Per assegnare a uno specifico indirizzo IP privato secondario un'interfaccia di rete

Questo esempio assegna l'indirizzo IP privato secondario specificato all'interfaccia di rete specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-ip-addresses 10.0.0.82
```

Per assegnare indirizzi IP privati secondari EC2 selezionati da Amazon a un'interfaccia di rete

Questo esempio assegna due indirizzi IP privati secondari all'interfaccia di rete specificata. Amazon assegna EC2 automaticamente questi indirizzi IP dagli indirizzi IP disponibili nell'intervallo di CIDR blocchi della sottorete a cui è associata l'interfaccia di rete. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --secondary-private-ip-address-count 2
```

- Per API i dettagli, consulta AWS CLI Command [AssignPrivateIpAddresses](#) Reference.

## assign-private-nat-gateway-address

Il seguente esempio di codice mostra come utilizzare `assign-private-nat-gateway-address`.

### AWS CLI

Per assegnare indirizzi IP privati al gateway privato NAT

L'esempio seguente assegna due indirizzi IP privati al gateway privato NAT specificato.

```
aws ec2 assign-private-nat-gateway-address \  
  --nat-gateway-id nat-1234567890abcdef0 \  
  --private-ip-address-count 2
```

**Output:**

```
{
  "NatGatewayId": "nat-1234567890abcdef0",
  "NatGatewayAddresses": [
    {
      "NetworkInterfaceId": "eni-0065a61b324d1897a",
      "IsPrimary": false,
      "Status": "assigning"
    },
    {
      "NetworkInterfaceId": "eni-0065a61b324d1897a",
      "IsPrimary": false,
      "Status": "assigning"
    }
  ]
}
```

Per ulteriori informazioni, consulta i [NATgateway](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [AssignPrivateNatGatewayAddress AWS CLI Command Reference](#).

**associate-address**

Il seguente esempio di codice mostra come utilizzare `associate-address`.

**AWS CLI**

Per associare un indirizzo IP elastico in EC2 -Classic

Questo esempio associa un indirizzo IP elastico a un'istanza in EC2 -Classic. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 associate-address --instance-id i-07ffe74c7330ebf53 --public-ip 198.51.100.0
```

Per associare un indirizzo IP elastico in - EC2 VPC

Questo esempio associa un indirizzo IP elastico a un'istanza in aVPC.

Comando:

```
aws ec2 associate-address --instance-id i-0b263919b6498b123 --allocation-id eipalloc-64d5890a
```

Output:

```
{
  "AssociationId": "eipassoc-2bebb745"
}
```

Nell'esempio seguente viene associato un indirizzo IP elastico a un'interfaccia di rete.

Comando:

```
aws ec2 associate-address --allocation-id eipalloc-64d5890a --network-interface-id eni-1a2b3c4d
```

Nell'esempio seguente viene associato un indirizzo IP elastico a un indirizzo IP privato associato a un'interfaccia di rete.

Comando:

```
aws ec2 associate-address --allocation-id eipalloc-64d5890a --network-interface-id eni-1a2b3c4d --private-ip-address 10.0.0.85
```

- Per API i dettagli, vedere [AssociateAddress](#) in AWS CLI Command Reference.

## **associate-client-vpn-target-network**

Il seguente esempio di codice mostra come utilizzare `associate-client-vpn-target-network`.

AWS CLI

Per associare una rete di destinazione a un VPN endpoint Client

L'esempio seguente associa una sottorete all'endpoint Client specificato. VPN

```
aws ec2 associate-client-vpn-target-network \
  --subnet-id subnet-0123456789abcabca \
```

```
--client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Output:

```
{
  "AssociationId": "cvpn-assoc-12312312312312312",
  "Status": {
    "Code": "associating"
  }
}
```

Per ulteriori informazioni, consulta [Target Networks](#) nella AWS Client VPN Administrator Guide.

- Per API i dettagli, vedere [AssociateClientVpnTargetNetwork](#) in AWS CLI Command Reference.

## associate-dhcp-options

Il seguente esempio di codice mostra come utilizzare `associate-dhcp-options`.

### AWS CLI

Per associare un set di DHCP opzioni al VPC

Questo esempio associa il set di DHCP opzioni specificato a quello specificato VPC. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 associate-dhcp-options --dhcp-options-id dopt-d9070ebb --vpc-id vpc-a01106c2
```

Per associare il set DHCP di opzioni predefinito al VPC

Questo esempio associa le DHCP opzioni predefinite impostate a quelle VPC specificate. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 associate-dhcp-options --dhcp-options-id default --vpc-id vpc-a01106c2
```

- Per API i dettagli, vedere [AssociateDhcpOptions](#) in AWS CLI Command Reference.

## associate-iam-instance-profile

Il seguente esempio di codice mostra come utilizzare `associate-iam-instance-profile`.

### AWS CLI

Per associare un profilo di IAM istanza a un'istanza

Questo esempio associa un profilo di IAM istanza denominato `admin-role` a un'istanza `i-123456789abcde123`.

Comando:

```
aws ec2 associate-iam-instance-profile --instance-id i-123456789abcde123 --iam-  
instance-profile Name=admin-role
```

Output:

```
{  
  "IamInstanceProfileAssociation": {  
    "InstanceId": "i-123456789abcde123",  
    "State": "associating",  
    "AssociationId": "iip-assoc-0e7736511a163c209",  
    "IamInstanceProfile": {  
      "Id": "AIPAJBLK7RKJKWDXVHIEC",  
      "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"  
    }  
  }  
}
```

- Per API i dettagli, vedere [AssociateIamInstanceProfile](#) in AWS CLI Command Reference.

## associate-instance-event-window

Il seguente esempio di codice mostra come utilizzare `associate-instance-event-window`.

### AWS CLI

Esempio 1: associare una o più istanze a una finestra di eventi

L'esempio seguente associa una o più istanze a una finestra degli eventi.

```
aws ec2 associate-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

Output:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "Name": "myEventWindowName",  
    "CronExpression": "* 21-23 * * 2,3",  
    "AssociationTarget": {  
      "InstanceIds": [  
        "i-1234567890abcdef0",  
        "i-0598c7d356eba48d7"  
      ],  
      "Tags": [],  
      "DedicatedHostIds": []  
    },  
    "State": "creating"  
  }  
}
```

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella sezione Eventi pianificati della Amazon EC2 User Guide.

Esempio 2: associare i tag di istanza a una finestra di eventi

L'associate-instance-event-windowesempio seguente associa i tag di istanza a una finestra di eventi. Immettete un instance-event-window-id parametro per specificare la finestra dell'evento. Per associare i tag di istanza, specificate il association-target parametro e, per il valore del parametro, specificate uno o più tag.

```
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}]"
```

Output:



```
{
  "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"
  }
}
```

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella sezione Eventi pianificati della Amazon EC2 User Guide.

Esempio 3: associare un host dedicato a una finestra di evento

L'associate-instance-event-windowesempio seguente associa un Dedicated Host a una finestra di evento. Immettete un `instance-event-window-id` parametro per specificare la finestra dell'evento. Per associare un host dedicato, specificare il `--association-target` parametro e, per i valori dei parametri, specificare uno o più host dedicati IDs.

```
aws ec2 associate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "DedicatedHostIds=h-029fa35a02b99801d"
```

Output:

```
{
```

```

    "InstanceEventWindow": {
      "InstanceEventWindowId": "iew-0abcdef1234567890",
      "Name": "myEventWindowName",
      "CronExpression": "* 21-23 * * 2,3",
      "AssociationTarget": {
        "InstanceIds": [],
        "Tags": [],
        "DedicatedHostIds": [
          "h-029fa35a02b99801d"
        ]
      },
      "State": "creating"
    }
  }
}

```

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella sezione Eventi pianificati della Amazon EC2 User Guide.

- Per API i dettagli, consulta Command [AssociateInstanceEventWindow](#) Reference AWS CLI .

## associate-ipam-resource-discovery

Il seguente esempio di codice mostra come utilizzare `associate-ipam-resource-discovery`.

### AWS CLI

Associare una scoperta di risorse a un IPAM

In questo esempio, l'utente è un amministratore IPAM delegato e un sistema di rilevamento delle risorse è stato creato e condiviso con l'utente da un altro AWS account, in modo da poterlo utilizzare IPAM per gestire e monitorare le risorse CIDRs di proprietà dell'altro account.

### Nota

Per completare questa richiesta, avrai bisogno dell'ID di scoperta della risorsa con cui puoi ottenere [describe-ipam-resource-discoveries](#) dell'IPAMID che puoi ottenere con [describe-ipams](#) .La scoperta della risorsa che stai associando deve essere stata prima condivisa con il tuo account utilizzando AWS RAM .La `--region` regione che inserisci deve corrispondere alla regione di origine a IPAM cui la stai associando.

L'esempio seguente `associate-ipam-resource-discovery` associa una scoperta di risorse a un. 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
```

Output:

```
{
  {
    "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": []
    }
  }
}
```

Dopo aver associato un rilevamento delle risorse, è possibile monitorare e/o gestire gli indirizzi IP delle risorse create dagli altri account. Per ulteriori informazioni, consulta [Integrazione IPAM con account esterni alla tua organizzazione](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [AssociateIpamResourceDiscovery AWS CLI Command Reference](#).

## associate-nat-gateway-address

Il seguente esempio di codice mostra come utilizzare `associate-nat-gateway-address`.

AWS CLI

Per associare un indirizzo IP elastico a un NAT gateway pubblico

L'associazione `associate-nat-gateway-address` seguente associa l'indirizzo IP elastico specificato al NAT gateway pubblico specificato. AWS assegna automaticamente un indirizzo privato IPv4 secondario.

```
aws ec2 associate-nat-gateway-address \
  --nat-gateway-id nat-1234567890abcdef0 \
  --allocation-ids eipalloc-0be6ecac95EXAMPLE
```

Output:

```
{
  "NatGatewayId": "nat-1234567890abcdef0",
  "NatGatewayAddresses": [
    {
      "AllocationId": "eipalloc-0be6ecac95EXAMPLE",
      "NetworkInterfaceId": "eni-09cc4b2558794f7f9",
      "IsPrimary": false,
      "Status": "associating"
    }
  ]
}
```

Per ulteriori informazioni, consulta i [NATgateway](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [AssociateNatGatewayAddress AWS CLI Command Reference](#).

## **associate-route-table**

Il seguente esempio di codice mostra come utilizzare `associate-route-table`.

AWS CLI

Per associare una tabella di routing a una sottorete

Questo esempio associa la tabella di routing specificata alla sottorete specificata.

Comando:

```
aws ec2 associate-route-table --route-table-id rtb-22574640 --subnet-
id subnet-9d4a7b6c
```

Output:

```
{
  "AssociationId": "rtbassoc-781d0d1a"
}
```

- Per API i dettagli, vedere [AssociateRouteTable](#) in AWS CLI Command Reference.

## associate-subnet-cidr-block

Il seguente esempio di codice mostra come utilizzare `associate-subnet-cidr-block`.

### AWS CLI

Per associare un IPv6 CIDR blocco a una sottorete

Questo esempio associa un IPv6 CIDR blocco alla sottorete specificata.

Comando:

```
aws ec2 associate-subnet-cidr-block --subnet-id subnet-5f46ec3b --ipv6-cidr-  
block 2001:db8:1234:1a00::/64
```

Output:

```
{
  "SubnetId": "subnet-5f46ec3b",
  "Ipv6CidrBlockAssociation": {
    "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
    "AssociationId": "subnet-cidr-assoc-3aa54053",
    "Ipv6CidrBlockState": {
      "State": "associating"
    }
  }
}
```

- Per API i dettagli, vedere [AssociateSubnetCidrBlock](#) in AWS CLI Command Reference.

## associate-transit-gateway-multicast-domain

Il seguente esempio di codice mostra come utilizzare `associate-transit-gateway-multicast-domain`.

## AWS CLI

Per associare un gateway di transito a un dominio multicast

L'esempio seguente associa la sottorete e l'allegato specificati al dominio multicast specificato.

```
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
```

Output:

```
{  
  "Associations": [  
    {  
      "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",  
      "TransitGatewayAttachmentId": "tgw-attach-028c1dd0f8f5cbe8e",  
      "ResourceId": "vpc-01128d2c240c09bd5",  
      "ResourceType": "vpc",  
      "Subnets": [  
        {  
          "SubnetId": "subnet-000de86e3b49c932a",  
          "State": "associating"  
        }  
      ]  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Managing multicast Domains](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [AssociateTransitGatewayMulticastDomain](#) in AWS CLI Command Reference.

## **associate-transit-gateway-route-table**

Il seguente esempio di codice mostra come utilizzare `associate-transit-gateway-route-table`.

## AWS CLI

Per associare una tabella di routing del gateway di transito a un allegato del gateway di transito

L'esempio seguente associa la tabella di routing del gateway di transito specificata all'VPC allegato specificato.

```
aws ec2 associate-transit-gateway-route-table \  
  --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \  
  --transit-gateway-attachment-id tgw-attach-0b5968d3b6EXAMPLE
```

Output:

```
{  
  "Association": {  
    "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",  
    "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",  
    "ResourceId": "vpc-0065acced4EXAMPLE",  
    "ResourceType": "vpc",  
    "State": "associating"  
  }  
}
```

Per ulteriori informazioni, consulta [Associare una tabella di routing del Transit Gateway](#) nella AWS Transit Gateways Guide.

- Per API i dettagli, vedere [AssociateTransitGatewayRouteTable](#) in AWS CLI Command Reference.

## associate-vpc-cidr-block

Il seguente esempio di codice mostra come utilizzare `associate-vpc-cidr-block`.

### AWS CLI

Esempio 1: associare un IPv6 CIDR blocco fornito da Amazon a un VPC

L'esempio seguente associa un IPv6 CIDR blocco al valore specificato. VPC :

```
aws ec2 associate-vpc-cidr-block \  
  --vpc-id vpc-0065acced4EXAMPLE \  
  --cidr-block 2001:db8:1:1::/64
```

```
--amazon-provided-ipv6-cidr-block \  
--ipv6-cidr-block-network-border-group us-west-2-lax-1 \  
--vpc-id vpc-8EXAMPLE
```

Output:

```
{  
  "Ipv6CidrBlockAssociation": {  
    "AssociationId": "vpc-cidr-assoc-0838ce7d9dEXAMPLE",  
    "Ipv6CidrBlockState": {  
      "State": "associating"  
    },  
    "NetworkBorderGroup": "us-west-2-lax-1"  
  },  
  "VpcId": "vpc-8EXAMPLE"  
}
```

Esempio 2: Per associare un IPv4 CIDR blocco aggiuntivo a VPC

L'associate-vpc-cidr-block esempio seguente associa il IPv4 CIDR blocco `10.2.0.0/16` al valore specificato. VPC

```
aws ec2 associate-vpc-cidr-block \  
--vpc-id vpc-1EXAMPLE \  
--cidr-block 10.2.0.0/16
```

Output:

```
{  
  "CidrBlockAssociation": {  
    "AssociationId": "vpc-cidr-assoc-2EXAMPLE",  
    "CidrBlock": "10.2.0.0/16",  
    "CidrBlockState": {  
      "State": "associating"  
    }  
  },  
  "VpcId": "vpc-1EXAMPLE"  
}
```

- Per API i dettagli, vedere [AssociateVpcCidrBlock](#) in AWS CLI Command Reference.



## attach-classic-link-vpc

Il seguente esempio di codice mostra come utilizzare `attach-classic-link-vpc`.

### AWS CLI

Per collegare (allegare) un'istanza EC2 -Classic a un VPC

Questo esempio collega l'istanza `i-1234567890abcdef0` a `vpc-88888888` tramite il gruppo di sicurezza `sg-12312312`. VPC VPC

Comando:

```
aws ec2 attach-classic-link-vpc --instance-id i-1234567890abcdef0 --vpc-id vpc-88888888 --groups sg-12312312
```

Output:

```
{  
  "Return": true  
}
```

- API Per i [AttachClassicLinkVpc](#) dettagli, vedere in AWS CLI Command Reference.

## attach-internet-gateway

Il seguente esempio di codice mostra come utilizzare `attach-internet-gateway`.

### AWS CLI

Per collegare un gateway Internet al tuo VPC

L'attach-internet-gateway esempio seguente collega il gateway Internet specificato allo specifico VPC.

```
aws ec2 attach-internet-gateway \  
  --internet-gateway-id igw-0d0fb496b3EXAMPLE \  
  --vpc-id vpc-0a60eb65b4EXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Internet gateway](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [AttachInternetGateway AWS CLI Command Reference](#).

## attach-network-interface

Il seguente esempio di codice mostra come utilizzare `attach-network-interface`.

### AWS CLI

Esempio 1: per collegare un'interfaccia di rete a un'istanza

L'`attach-network-interface` esempio seguente collega l'interfaccia di rete specificata all'istanza specificata.

```
aws ec2 attach-network-interface \  
  --network-interface-id eni-0dc56a8d4640ad10a \  
  --instance-id i-1234567890abcdef0 \  
  --device-index 1
```

Output:

```
{  
  "AttachmentId": "eni-attach-01a8fc87363f07cf9"  
}
```

Per ulteriori informazioni, consulta [Interfacce di rete elastiche](#) nella Amazon EC2 User Guide.

Esempio 2: collegare un'interfaccia di rete a un'istanza con più schede di rete

L'`attach-network-interface` esempio seguente collega l'interfaccia di rete specificata all'istanza e alla scheda di rete specificate.

```
aws ec2 attach-network-interface \  
  --network-interface-id eni-07483b1897541ad83 \  
  --instance-id i-01234567890abcdef \  
  --network-card-index 1 \  
  --device-index 1
```

Output:

```
{
  "AttachmentId": "eni-attach-0fbd7ee87a88cd06c"
}
```

Per ulteriori informazioni, consulta [Interfacce di rete elastiche](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [AttachNetworkInterface AWS CLI Command Reference](#).

## attach-verified-access-trust-provider

Il seguente esempio di codice mostra come utilizzare `attach-verified-access-trust-provider`.

### AWS CLI

Per collegare un provider di fiducia a un'istanza

L'`attach-verified-access-trust-provider` seguente collega il provider fiduciario di accesso verificato specificato all'istanza di accesso verificato specificata.

```
aws ec2 attach-verified-access-trust-provider \
  --verified-access-instance-id vai-0ce000c0b7643abea \
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

Output:

```
{
  "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"
}
}

```

Per ulteriori informazioni, consulta le [istanze di accesso verificato nella Guida](#) per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [AttachVerifiedAccessTrustProvider AWS CLI Command Reference](#).

## attach-volume

Il seguente esempio di codice mostra come utilizzare `attach-volume`.

### AWS CLI

Per collegare un volume a un'istanza

Questo comando di esempio collega un volume (`vol-1234567890abcdef0`) a un'istanza (`i-01474ef662b89480`) come `/dev/sdf`.

Comando:

```
aws ec2 attach-volume --volume-id vol-1234567890abcdef0 --instance-id i-01474ef662b89480 --device /dev/sdf
```

Output:

```
{
  "AttachTime": "YYYY-MM-DDTHH:MM:SS.000Z",
  "InstanceId": "i-01474ef662b89480",
  "VolumeId": "vol-1234567890abcdef0",
  "State": "attaching",
  "Device": "/dev/sdf"
}
```

- Per API i dettagli, vedere [AttachVolume](#) in AWS CLI Command Reference.

## attach-vpn-gateway

Il seguente esempio di codice mostra come utilizzare `attach-vpn-gateway`.

### AWS CLI

Per collegare un gateway privato virtuale al tuo VPC

L'esempio seguente collega il gateway privato virtuale specificato a quello specificato VPC.

```
aws ec2 attach-vpn-gateway \  
  --vpn-gateway-id vgw-9a4cacf3 \  
  --vpc-id vpc-a01106c2
```

Output:

```
{  
  "VpcAttachment": {  
    "State": "attaching",  
    "VpcId": "vpc-a01106c2"  
  }  
}
```

- Per API i dettagli, vedere [AttachVpnGateway](#) in AWS CLI Command Reference.

## authorize-client-vpn-ingress

Il seguente esempio di codice mostra come utilizzare `authorize-client-vpn-ingress`.

### AWS CLI

Per aggiungere una regola di autorizzazione per un VPN endpoint Client

L'esempio seguente aggiunge una regola di autorizzazione all'ingresso che consente a tutti i clienti di accedere a Internet (`0.0.0.0/0`).

```
aws ec2 authorize-client-vpn-ingress \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --target-network-cidr 0.0.0.0/0 \  
  --authorize-all-groups
```

**Output:**

```
{
  "Status": {
    "Code": "authorizing"
  }
}
```

Per ulteriori informazioni, vedere [Regole di autorizzazione](#) nella AWS Client VPN Administrator Guide.

- Per API i dettagli, vedere [AuthorizeClientVpnIngress](#) in AWS CLI Command Reference.

**authorize-security-group-egress**

Il seguente esempio di codice mostra come utilizzare `authorize-security-group-egress`.

**AWS CLI**

Per aggiungere una regola che consenta il traffico in uscita verso un intervallo di indirizzi specifico

Questo comando di esempio aggiunge una regola che concede l'accesso agli intervalli di indirizzi specificati sulla TCP porta 80.

Comando (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}]'
```

Comando (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}]
```

Per aggiungere una regola che consenta il traffico in uscita verso un gruppo di sicurezza specifico

Questo comando di esempio aggiunge una regola che concede l'accesso al gruppo di sicurezza specificato sulla TCP porta 80.

Comando (Linux):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-permissions  
IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=' [{GroupId=sg-4b51a32f}]'
```

Comando (Windows):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-  
permissions IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=[{GroupId=sg-4b51a32f}]
```

- Per API i dettagli, vedere [AuthorizeSecurityGroupEgress](#) in AWS CLI Command Reference.

## authorize-security-group-ingress

Il seguente esempio di codice mostra come utilizzare `authorize-security-group-ingress`.

### AWS CLI

Esempio 1: aggiungere una regola che consenta il traffico in entrata SSH

L'`authorize-security-group-ingress` seguente aggiunge una regola che consente il traffico in entrata sulla TCP porta 22 (SSH).

```
aws ec2 authorize-security-group-ingress \  
  --group-id sg-1234567890abcdef0 \  
  --protocol tcp \  
  --port 22 \  
  --cidr 203.0.113.0/24
```

Output:

```
{  
  "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"
  }
]
}

```

Esempio 2: Per aggiungere una regola che consenta il HTTP traffico in entrata da un altro gruppo di sicurezza

L'`authorize-security-group-ingress` seguente aggiunge una regola che consente l'accesso in entrata sulla TCP porta 80 dal gruppo di sicurezza di origine. `sg-1a2b3c4d` Il gruppo di origine deve appartenere allo stesso gruppo VPC o a un peer VPC (richiede una connessione VPC peering). Il traffico in entrata è autorizzato in base agli indirizzi IP privati delle istanze associate al gruppo di sicurezza di origine (e non in base all'indirizzo IP elastico o pubblico).

```

aws ec2 authorize-security-group-ingress \
  --group-id sg-1234567890abcdef0 \
  --protocol tcp \
  --port 80 \
  --source-group sg-1a2b3c4d

```

Output:

```

{
  "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"
      }
    }
  ]
}

```



### Esempio 3: per aggiungere più regole nella stessa chiamata

L'`authorize-security-group-ingress` seguente utilizza il `ip-permissions` parametro per aggiungere due regole in entrata, una che abilita l'accesso in entrata sulla TCP porta 3389 (RDP) e l'altra che abilita ping/. 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}]"
```

Output:

```
{
  "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"
    }
  ]
}
```

### Esempio 4: aggiungere una ICMP regola per il traffico

L'`authorize-security-group-ingress` seguente utilizza il `ip-permissions` parametro per aggiungere una regola in entrata che consente il ICMP messaggio Destination

Unreachable: Fragmentation Needed and Don't Fragment was Set (Tipo 3, Codice 4) da qualsiasi luogo.

```
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
```

Output:

```
{
  "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"
    }
  ]
}
```

Esempio 5: aggiungere una regola per il IPv6 traffico

L'authorize-security-group-ingress seguente utilizza il ip-permissions parametro per aggiungere una regola in entrata che consente SSH l'accesso (porta 22) dall'IPv6 intervallo 2001:db8:1234:1a00::/64.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
=tcp, =22, =22, Ipv6Ranges= [{"6=2001:db 8:1234:1 a00: IpProtocol :/64}]]» FromPort ToPort
CidrIpV
```

Output:

```
{
  "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"
  }
]
}

```

### Esempio 6: aggiungere una regola per ICMPv6 il traffico

L'`authorize-security-group-ingress` seguente utilizza il `ip-permissions` parametro per aggiungere una regola in entrata che consente il ICMPv6 traffico da qualsiasi luogo.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
=icmpv6, Ipv6Ranges="[{6=: :0}]» IpProtocol CidrIpv6
```

Output:

```

{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-04b612d9363ab6327",
      "GroupId": "sg-1234567890abcdef0",
      "GroupOwnerId": "123456789012",
      "IsEgress": false,
      "IpProtocol": "icmpv6",
      "FromPort": -1,
      "ToPort": -1,
      "CidrIpv6": "::/0"
    }
  ]
}

```

### Esempio 7: aggiungere una regola con una descrizione

L'`authorize-security-group-ingress` seguente utilizza il parametro per aggiungere una regola in entrata `ip-permissions` che consente il traffico dall'intervallo di

indirizzi specificato. RDP IPv4 La regola include una descrizione per consentirne l'identificazione in un secondo momento.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions
IpProtocol=tcp,=3389,=3389,=3389,="[=203.0.113.0/24, Description='accesso dall'ufficio di
NY']» FromPort ToPort IpRanges CidrIp RDP
```

Output:

```
{
  "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"
    }
  ]
}
```

Esempio 8: per aggiungere una regola in entrata che utilizza un elenco di prefissi

L'`authorize-security-group-ingress` seguente utilizza il parametro `ip-permissions` per aggiungere una regola in entrata che consente tutto il traffico per gli intervalli nell'elenco di prefissi CIDR specificato.

```
aws ec2 authorize-security-group-ingress --group-id sg-04a351bfe432d4e71 --ip-permissions=all,
="[=pl-002dc3ec097de1514]» IpProtocol PrefixListIds PrefixListId
```

Output:

```
{
  "Return": true,
  "SecurityGroupRules": [
    {
```

```

        "SecurityGroupRuleId": "sgr-09c74b32f677c6c7c",
        "GroupId": "sg-1234567890abcdef0",
        "GroupOwnerId": "123456789012",
        "IsEgress": false,
        "IpProtocol": "-1",
        "FromPort": -1,
        "ToPort": -1,
        "PrefixListId": "pl-0721453c7ac4ec009"
    }
]
}

```

Per ulteriori informazioni, consulta [Security groups](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [AuthorizeSecurityGroupIngress AWS CLI Command Reference](#).

## bundle-instance

Il seguente esempio di codice mostra come utilizzare `bundle-instance`.

### AWS CLI

Per raggruppare un'istanza

Questo esempio raggruppa l'istanza `i-1234567890abcdef0` in un bucket chiamato.

`bundletasks` Prima di specificare i valori per la chiave di accesso IDs, consulta e segui le indicazioni contenute nelle migliori pratiche per la gestione delle chiavi di AWS accesso.

Comando:

```
aws ec2 bundle-instance --instance-id i-1234567890abcdef0 --bucket bundletasks --
prefix winami --owner-akid AK12AJEXAMPLE --owner-sak example123example
```

Output:

```

{
  "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"
}
}
```

- Per API i dettagli, consulta [BundleInstance AWS CLI Command Reference](#).

## cancel-bundle-task

Il seguente esempio di codice mostra come utilizzare `cancel-bundle-task`.

### AWS CLI

Per annullare un'operazione associata a un pacchetto

Questo esempio annulla l'operazione di raggruppamento. `bun-2a4e041c`

Comando:

```
aws ec2 cancel-bundle-task --bundle-id bun-2a4e041c
```

Output:

```
{
  "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"
  }
}
```

```
}  
}
```

- Per API i dettagli, vedere [CancelBundleTask](#) in AWS CLI Command Reference.

## cancel-capacity-reservation-fleets

Il seguente esempio di codice mostra come utilizzare `cancel-capacity-reservation-fleets`.

### AWS CLI

Per annullare una flotta di prenotazioni di capacità

L'esempio seguente annulla la flotta di prenotazione della capacità specificata e la capacità che riserva. Quando si annulla una flotta, il suo stato cambia e non è più possibile creare nuove prenotazioni di capacità. Inoltre, tutte le prenotazioni di capacità individuali del parco veicoli vengono annullate e le istanze precedentemente in esecuzione con capacità riservata continuano a funzionare normalmente con capacità condivisa.

```
aws ec2 cancel-capacity-reservation-fleets \  
  --capacity-reservation-fleet-ids crf-abcdef01234567890
```

Output:

```
{  
  "SuccessfulFleetCancellations": [  
    {  
      "CurrentFleetState": "cancelling",  
      "PreviousFleetState": "active",  
      "CapacityReservationFleetId": "crf-abcdef01234567890"  
    }  
  ],  
  "FailedFleetCancellations": []  
}
```

Per ulteriori informazioni sulle flotte di prenotazione della capacità, consulta [Flotte di prenotazione della capacità nella Guida](#) per gli EC2 utenti di Amazon.

- Per API i dettagli, consulta AWS CLI Command [CancelCapacityReservationFleets](#) Reference.

## cancel-capacity-reservation

Il seguente esempio di codice mostra come utilizzare `cancel-capacity-reservation`.

### AWS CLI

Per annullare una prenotazione di capacità

L'esempio seguente annulla la prenotazione di capacità specificata.

```
aws ec2 cancel-capacity-reservation \
  --capacity-reservation-id cr-1234abcd56EXAMPLE
```

Output:

```
{
  "Return": true
}
```

Per ulteriori informazioni, consulta [Annullamento di una prenotazione di capacità](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per i API dettagli, consulta [CancelCapacityReservation](#) Command Reference.AWS CLI

## cancel-conversion-task

Il seguente esempio di codice mostra come utilizzare `cancel-conversion-task`.

### AWS CLI

Per annullare una conversione attiva di un'istanza o di un volume

Questo esempio annulla il caricamento associato all'ID attività `import-i-fh 95npoc`. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 cancel-conversion-task --conversion-task-id import-i-fh95npoc
```

- Per i API dettagli, consulta [CancelConversionTask](#) Command Reference.AWS CLI



## cancel-export-task

Il seguente esempio di codice mostra come utilizzare `cancel-export-task`.

### AWS CLI

Per annullare un'attività di esportazione attiva

Questo esempio annulla un'attività di esportazione attiva con l'ID attività `export-i-fgelt0i7`. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 cancel-export-task --export-task-id export-i-fgelt0i7
```

- Per API i dettagli, vedere [CancelExportTask](#) in AWS CLI Command Reference.

## cancel-image-launch-permission

Il seguente esempio di codice mostra come utilizzare `cancel-image-launch-permission`.

### AWS CLI

Per annullare la AMI condivisione di un account con il tuo account Amazon Web Services

L'`cancel-image-launch-permission` esempio seguente rimuove il tuo account dalle autorizzazioni AMI di avvio specificate.

```
aws ec2 cancel-image-launch-permission \  
  --image-id ami-0123456789example \  
  --region us-east-1
```

Output:

```
{  
  "Return": true  
}
```

Per ulteriori informazioni, consulta [Annullare la AMI condivisione con il tuo account Amazon Web Services](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [CancelImageLaunchPermission AWS CLI Command Reference](#).

## cancel-import-task

Il seguente esempio di codice mostra come utilizzare `cancel-import-task`.

### AWS CLI

Per annullare un'operazione di importazione

L'esempio seguente annulla l'attività di importazione dell'immagine specificata.

```
aws ec2 cancel-import-task \  
  --import-task-id import-ami-1234567890abcdef0
```

Output:

```
{  
  "ImportTaskId": "import-ami-1234567890abcdef0",  
  "PreviousState": "active",  
  "State": "deleting"  
}
```

- Per API i dettagli, vedere [CancelImportTask](#) in AWS CLI Command Reference.

## cancel-reserved-instances-listing

Il seguente esempio di codice mostra come utilizzare `cancel-reserved-instances-listing`.

### AWS CLI

Per annullare un elenco di istanze riservate

L'esempio seguente annulla l'elenco delle istanze riservate specificato.

```
aws ec2 cancel-reserved-instances-listing \  
  --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

- Per API i dettagli, vedere [CancelReservedInstancesListing](#) in AWS CLI Command Reference.

## cancel-spot-fleet-requests

Il seguente esempio di codice mostra come utilizzare `cancel-spot-fleet-requests`.

### AWS CLI

Esempio 1: annullare una richiesta del parco istanze Spot e terminare le istanze associate

L'`cancel-spot-fleet-requests` esempio seguente annulla una richiesta Spot Fleet e termina le istanze On-Demand e le istanze Spot associate.

```
aws ec2 cancel-spot-fleet-requests \  
  --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
  --terminate-instances
```

Output:

```
{  
  "SuccessfulFleetRequests": [  
    {  
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",  
      "CurrentSpotFleetRequestState": "cancelled_terminating",  
      "PreviousSpotFleetRequestState": "active"  
    }  
  ],  
  "UnsuccessfulFleetRequests": []  
}
```

Per ulteriori informazioni, consulta [Annullare una richiesta Spot Fleet](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

Esempio 2: annullare una richiesta del parco istanze Spot senza terminare le istanze associate

L'`cancel-spot-fleet-requests` esempio seguente annulla una richiesta Spot Fleet senza terminare le istanze On-Demand e le istanze Spot associate.

```
aws ec2 cancel-spot-fleet-requests \  
  --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \  
  --no-terminate-instances
```

```
--no-terminate-instances
```

Output:

```
{
  "SuccessfulFleetRequests": [
    {
      "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
      "CurrentSpotFleetRequestState": "cancelled_running",
      "PreviousSpotFleetRequestState": "active"
    }
  ],
  "UnsuccessfulFleetRequests": []
}
```

Per ulteriori informazioni, consulta [Annullare una richiesta Spot Fleet](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

- Per API i dettagli, consulta AWS CLI Command [CancelSpotFleetRequests](#) Reference.

## **cancel-spot-instance-requests**

Il seguente esempio di codice mostra come utilizzare `cancel-spot-instance-requests`.

AWS CLI

Per annullare le richieste di istanze Spot

Questo comando di esempio annulla una richiesta di istanza Spot.

Comando:

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

Output:

```
{
  "CancelledSpotInstanceRequests": [
    {
      "State": "cancelled",
      "SpotInstanceRequestId": "sir-08b93456"
    }
  ]
}
```

```
    ]
  }
}
```

- Per API i dettagli, consulta [CancelSpotInstanceRequests AWS CLI Command Reference](#).

## confirm-product-instance

Il seguente esempio di codice mostra come utilizzare `confirm-product-instance`.

### AWS CLI

Per confermare l'istanza del prodotto

Questo esempio determina se il codice prodotto specificato è associato all'istanza specificata.

Comando:

```
aws ec2 confirm-product-instance --product-code 774F4FF8 --instance-  
id i-1234567890abcdef0
```

Output:

```
{  
  "OwnerId": "123456789012"  
}
```

- Per API i dettagli, vedere [ConfirmProductInstance](#) in AWS CLI Command Reference.

## copy-fpga-image

Il seguente esempio di codice mostra come utilizzare `copy-fpga-image`.

### AWS CLI

Per copiare un'FPGA immagine Amazon

Questo esempio copia il valore specificato AFI dalla `us-east-1` regione alla regione corrente (`eu-west-1`).

Comando:

```
aws ec2 copy-fpga-image --name copy-afi --source-fpga-image-id afi-0d123e123bfc85abc
--source-region us-east-1 --region eu-west-1
```

Output:

```
{
  "FpgaImageId": "afi-06b12350a123fbabc"
}
```

- Per API i dettagli, vedere [CopyFpgaImage](#) in AWS CLI Command Reference.

## copy-image

Il seguente esempio di codice mostra come utilizzare `copy-image`.

### AWS CLI

Esempio 1: copiare un file AMI in un'altra regione

Il comando di `copy-image` esempio seguente copia quanto specificato AMI dalla `us-west-2` Regione alla `us-east-1` Regione e aggiunge una breve descrizione.

```
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."
```

Output:

```
{
  "ImageId": "ami-0123456789abcdefg"
}
```

Per ulteriori informazioni, consulta [Copiare un file AMI](#) nella Amazon EC2 User Guide.

Esempio 2: copiare un file AMI in un'altra regione e crittografare lo snapshot di supporto

Il `copy-image` comando seguente copia l'istantanea specificata AMI dalla `us-west-2` regione alla regione corrente e crittografa l'istantanea di backup utilizzando la chiave specificata. KMS

```
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
```

Output:

```
{  
  "ImageId": "ami-0123456789abcdefg"  
}
```

Per ulteriori informazioni, consulta [Copiare un file AMI](#) nella Amazon EC2 User Guide.

Esempio 3: per includere i AMI tag definiti dall'utente quando si copia un AMI

Il `copy-image` comando seguente utilizza il `--copy-image-tags` parametro per copiare i AMI tag definiti dall'utente durante la copia di. 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
```

Output:

```
{  
  "ImageId": "ami-0123456789abcdefg"  
}
```

Per ulteriori informazioni, consulta [Copiare un file AMI](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [CopyImage AWS CLI Command Reference](#).

## copy-snapshot

Il seguente esempio di codice mostra come utilizzare `copy-snapshot`.

### AWS CLI

Esempio 1: copiare un'istantanea in un'altra regione

Il comando di `copy-snapshot` esempio seguente copia l'istantanea specificata dalla `us-west-2` Regione alla `us-east-1` Regione e aggiunge una breve descrizione.

```
aws ec2 copy-snapshot \  
  --region us-east-1 \  
  --source-region us-west-2 \  
  --source-snapshot-id snap-066877671789bd71b \  
  --description "This is my copied snapshot."
```

Output:

```
{  
  "SnapshotId": "snap-066877671789bd71b"  
}
```

Per ulteriori informazioni, consulta [Copiare uno EBS snapshot Amazon](#) nella Amazon EC2 User Guide.

Esempio 2: copiare uno snapshot non crittografato e crittografare il nuovo snapshot

Il `copy-snapshot` comando seguente copia l'istantanea non crittografata specificata dalla `us-west-2` regione alla regione corrente e crittografa la nuova istantanea utilizzando la chiave specificata. KMS

```
aws ec2 copy-snapshot \  
  --source-region us-west-2 \  
  --source-snapshot-id snap-066877671789bd71b \  
  --encrypted \  
  --kms-key-id alias/my-kms-key
```

Output:

```
{
```



```
"SnapshotId": "snap-066877671789bd71b"
}
```

Per ulteriori informazioni, consulta [Copiare uno EBS snapshot Amazon](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [CopySnapshot AWS CLI Command Reference](#).

## create-capacity-reservation-fleet

Il seguente esempio di codice mostra come utilizzare `create-capacity-reservation-fleet`.

### AWS CLI

Per creare una flotta di prenotazioni di capacità

L'esempio seguente crea una flotta di prenotazioni di capacità per il tipo di istanza specificato nella richiesta, fino alla capacità target totale specificata. Il numero di istanze per le quali il parco istanze di prenotazione della capacità prenota quest'ultima dipende dalla capacità target totale e dai pesi del tipo di istanza specificati nella richiesta. Specificate i tipi di istanza da utilizzare e una priorità per ciascuno dei tipi di istanza designati.

```
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
```

Contenuto di `instanceTypeSpecification.json`.

```
[
  {
    "InstanceType": "m5.xlarge",
    "InstancePlatform": "Linux/UNIX",
    "Weight": 3.0,
    "AvailabilityZone": "us-east-1a",
    "EbsOptimized": true,
    "Priority" : 1
  }
]
```

```
]
```

Output:

```
{
  "Status": "submitted",
  "TotalFulfilledCapacity": 0.0,
  "CapacityReservationFleetId": "crf-abcdef01234567890",
  "TotalTargetCapacity": 24
}
```

Per ulteriori informazioni sulle flotte di prenotazione della capacità, consulta [Flotte di prenotazione della capacità nella Guida](#) per gli EC2 utenti di Amazon.

Per ulteriori informazioni sul peso del tipo di istanza e sulla capacità target totale, consulta [Peso del tipo di istanza](#) e [Capacità target totale](#) nella Amazon EC2 User Guide.

Per ulteriori informazioni sulla designazione della priorità per tipi di istanze specifici, consulta [Strategia di allocazione](#) e [Priorità del tipo di istanza](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateCapacityReservationFleet](#) Reference.

## create-capacity-reservation

Il seguente esempio di codice mostra come utilizzare `create-capacity-reservation`.

AWS CLI

Esempio 1: creare una prenotazione di capacità

L'esempio seguente crea una prenotazione di capacità nella zona di `eu-west-1a` disponibilità, nella quale è possibile avviare tre `t2.medium` istanze che eseguono un sistema operativo Linux/Unix. Per impostazione predefinita, la prenotazione della capacità viene creata con criteri di corrispondenza delle istanze aperte e non supporta lo storage temporaneo e rimane attiva finché non viene annullata manualmente.

```
aws ec2 create-capacity-reservation \
  --availability-zone eu-west-1a \
  --instance-type t2.medium \
  --instance-platform Linux/UNIX \
  --instance-count 3
```

## Output:

```
{
  "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"
  }
}
```

Esempio 2: creare una prenotazione di capacità che termini automaticamente a una data/ora specificata

L'create-capacity-reservationesempio seguente crea una prenotazione di capacità nella zona di eu-west-1a disponibilità, nella quale è possibile avviare tre m5.large istanze che eseguono un sistema operativo Linux/Unix. Questa prenotazione di capacità termina automaticamente il 31/08/2019 alle 23:59:59.

```
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
```

## Output:

```
{
  "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"
  }
}

```

Esempio 3: creare una prenotazione di capacità che accetti solo lanci di istanze mirati

L'create-capacity-reservationesempio seguente crea una prenotazione di capacità che accetta solo avvii di istanze mirati.

```

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

```

Output:

```

{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Creazione di una prenotazione di capacità](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per API i dettagli, consulta AWS CLI Command [CreateCapacityReservationReference](#).

## create-carrier-gateway

Il seguente esempio di codice mostra come utilizzare `create-carrier-gateway`.

### AWS CLI

Per creare un carrier gateway

L'esempio seguente crea un carrier gateway per quanto specificato VPC.

```
aws ec2 create-carrier-gateway \
  --vpc-id vpc-0c529aEXAMPLE1111
```

Output:

```
{
  "CarrierGateway": {
    "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
    "VpcId": "vpc-0c529aEXAMPLE1111",
    "State": "pending",
    "OwnerId": "123456789012"
  }
}
```

Per ulteriori informazioni, consulta [Carrier Gateway](#) nella AWS Wavelength User Guide.

- Per API i dettagli, vedere [CreateCarrierGateway](#) in AWS CLI Command Reference.

## create-client-vpn-endpoint

Il seguente esempio di codice mostra come utilizzare `create-client-vpn-endpoint`.

## AWS CLI

Per creare un VPN endpoint Client

L'`create-client-vpn-endpoint` seguente crea un VPN endpoint Client che utilizza l'autenticazione reciproca e specifica un valore per il blocco client. CIDR

```
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
```

Output:

```
{
  "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
  "Status": {
    "Code": "pending-associate"
  },
  "DnsName": "cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-
south-1.amazonaws.com"
}
```

Per ulteriori informazioni, consulta [Client VPN Endpoints nella Client Administrator AWS Guide](#).  
VPN

- Per API i dettagli, vedere [CreateClientVpnEndpoint](#) in AWS CLI Command Reference.

## `create-client-vpn-route`

Il seguente esempio di codice mostra come utilizzare `create-client-vpn-route`.

## AWS CLI

Per creare un percorso per un VPN endpoint Client

L'`create-client-vpn-route` seguente aggiunge una route a internet (0.0.0.0/0) per la sottorete specificata dell'endpoint ClientVPN.

```
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
```

Output:

```
{  
  "Status": {  
    "Code": "creating"  
  }  
}
```

Per ulteriori informazioni, vedere [Routes](#) nella AWS Client VPN Administrator Guide.

- Per API i dettagli, vedere [CreateClientVpnRoute](#) in AWS CLI Command Reference.

## create-coip-cidr

Il seguente esempio di codice mostra come utilizzare `create-coip-cidr`.

### AWS CLI

Per creare un intervallo di indirizzi IP (CoIP) di proprietà del cliente

L'`create-coip-cidr` seguente crea l'intervallo specificato di indirizzi CoIP nel pool CoIP specificato.

```
aws ec2 create-coip-cidr \  
  --cidr 15.0.0.0/24 \  
  --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

Output:

```
{  
  "CoipCidr": {  
    "Cidr": "15.0.0.0/24",  
    "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",  
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"  
  }  
}
```

```
}
```

Per ulteriori informazioni, consulta [Indirizzi IP di proprietà del cliente](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, vedere [CreateCoipCidr](#) in AWS CLI Command Reference.

## create-coip-pool

Il seguente esempio di codice mostra come utilizzare `create-coip-pool`.

### AWS CLI

Per creare un pool di indirizzi IP (CoIP) di proprietà del cliente

L'esempio seguente crea un pool CoIP per gli indirizzi CoIP nella tabella di routing del gateway locale specificata.

```
aws ec2 create-coip-pool \
  --local-gateway-route-table-id lgw-rtb-abcdefg1234567890
```

Output:

```
{
  "CoipPool": {
    "PoolId": "ipv4pool-coip-1234567890abcdefg",
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
    "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-
coip-1234567890abcdefg"
  }
}
```

Per ulteriori informazioni, consulta [Indirizzi IP di proprietà del cliente](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, vedere [CreateCoipPool](#) in AWS CLI Command Reference.

## create-customer-gateway

Il seguente esempio di codice mostra come utilizzare `create-customer-gateway`.



## AWS CLI

Per creare un gateway per i clienti

Questo esempio crea un gateway per i clienti con l'indirizzo IP specificato per la relativa interfaccia esterna.

Comando:

```
aws ec2 create-customer-gateway --type ipsec.1 --public-ip 12.1.2.3 --bgp-asn 65534
```

Output:

```
{
  "CustomerGateway": {
    "CustomerGatewayId": "cgw-0e11f167",
    "IpAddress": "12.1.2.3",
    "State": "available",
    "Type": "ipsec.1",
    "BgpAsn": "65534"
  }
}
```

- Per API i dettagli, vedere [CreateCustomerGateway](#) in AWS CLI Command Reference.

## **create-default-subnet**

Il seguente esempio di codice mostra come utilizzare `create-default-subnet`.

## AWS CLI

Per creare una sottorete predefinita

Questo esempio crea una sottorete predefinita nella zona di disponibilità. `us-east-2a`

Comando:

```
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
}
```

- Per API i dettagli, vedere [CreateDefaultSubnet](#) in AWS CLI Command Reference.

## create-default-vpc

Il seguente esempio di codice mostra come utilizzare `create-default-vpc`.

### AWS CLI

Per creare un valore predefinito VPC

Questo esempio crea un valore predefinito VPC.

Comando:

```
aws ec2 create-default-vpc
```

Output:

```
{
  "Vpc": {
    "VpcId": "vpc-8eaae5ea",
    "InstanceTenancy": "default",
    "Tags": [],
    "Ipv6CidrBlockAssociationSet": [],
    "State": "pending",
    "DhcpOptionsId": "dopt-af0c32c6",
    "CidrBlock": "172.31.0.0/16",
```

```
    "IsDefault": true
  }
}
```

- Per API i dettagli, vedere [CreateDefaultVpc](#) in AWS CLI Command Reference.

## create-dhcp-options

Il seguente esempio di codice mostra come utilizzare `create-dhcp-options`.

### AWS CLI

Per creare un set di DHCP opzioni

L'`create-dhcp-options` esempio seguente crea un set di DHCP opzioni che specifica il nome di dominio, i server dei nomi di dominio e il tipo di BIOS nodo di rete.

```
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"
```

Output:

```
{  
  "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"
}
}

```

- Per API i dettagli, vedere [CreateDhcpOptions](#) in AWS CLI Command Reference.

## create-egress-only-internet-gateway

Il seguente esempio di codice mostra come utilizzare `create-egress-only-internet-gateway`.

### AWS CLI

Per creare un gateway Internet solo in uscita

Questo esempio crea un gateway Internet di sola uscita per l'area specificata. VPC

Comando:

```
aws ec2 create-egress-only-internet-gateway --vpc-id vpc-0c62a468
```

Output:

```

{
  "EgressOnlyInternetGateway": {
    "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
    "Attachments": [
      {
        "State": "attached",

```

```

    "VpcId": "vpc-0c62a468"
  }
]
}
}

```

- Per API i dettagli, vedere [CreateEgressOnlyInternetGateway](#) in AWS CLI Command Reference.

## create-fleet

Il seguente esempio di codice mostra come utilizzare `create-fleet`.

### AWS CLI

Esempio 1: creare una EC2 flotta che avvii le istanze Spot come modello di acquisto predefinito

L'esempio seguente crea una EC2 flotta utilizzando i parametri minimi richiesti per lanciare una flotta: un modello di lancio, una capacità target e un modello di acquisto predefinito. Il modello di avvio viene identificato dall'ID e dal numero di versione del modello di avvio. La capacità target per il parco istanze è di 2 istanze, mentre il modello di acquisto predefinito è `spot`, il che comporta il lancio di 2 istanze Spot da parte del parco istanze.

Quando crei un EC2 parco istanze, utilizza un JSON file per specificare le informazioni sulle istanze da avviare.

```

aws ec2 create-fleet \
  --cli-input-json file:///file_name.json

```

Contenuto di `file_name.json`:

```

{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    }
  ],
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,

```

```
    "DefaultTargetCapacityType": "spot"
  }
}
```

Output:

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

Esempio 2: creare una EC2 flotta che avvii le istanze On-Demand come modello di acquisto predefinito

L'`create-fleet` seguente crea una EC2 flotta utilizzando i parametri minimi richiesti per lanciare una flotta: un modello di lancio, una capacità target e un modello di acquisto predefinito. Il modello di avvio viene identificato dall'ID e dal numero di versione del modello di avvio. La capacità target per il parco istanze è di 2 istanze, mentre il modello di acquisto predefinito è on-demand, il che comporta il lancio di 2 istanze On-Demand da parte del parco istanze.

Quando crei un EC2 parco istanze, utilizza un JSON file per specificare le informazioni sulle istanze da avviare.

```
aws ec2 create-fleet \
  --cli-input-json file://file_name.json
```

Contenuto di `file_name.json`:

```
{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    }
  ],
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "DefaultTargetCapacityType": "on-demand"
  }
}
```

```
}
```

### Output:

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

### Esempio 3: creare una EC2 flotta che avvii istanze On-Demand come capacità principale

L'`create-fleet` seguente crea una EC2 flotta che specifica la capacità target totale di 2 istanze per il parco istanze e una capacità target di 1 istanza On-Demand. Il modello di acquisto predefinito è `spot`. Il parco istanze lancia 1 istanza On-Demand come specificato, ma deve avviare un'altra istanza per soddisfare la capacità totale prevista. Il modello di acquisto per la differenza viene calcolato come `TotalTargetCapacity - OnDemandTargetCapacity = DefaultTargetCapacityType`, il che comporta il lancio di 1 istanza Spot da parte del parco macchine.

Quando crei un EC2 parco istanze, utilizza un JSON file per specificare le informazioni sulle istanze da avviare.

```
aws ec2 create-fleet \
  --cli-input-json file://file_name.json
```

### Contenuto di `file_name.json`:

```
{
  "LaunchTemplateConfigs": [
    {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
      }
    }
  ],
  "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "OnDemandTargetCapacity": 1,
    "DefaultTargetCapacityType": "spot"
  }
}
```

```
}
```

Output:

```
{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

Esempio 4: creare una EC2 flotta che lanci istanze Spot utilizzando la strategia di allocazione al prezzo più basso

Se non è indicata la strategia di allocazione per le Istanze spot, viene utilizzata la strategia di allocazione predefinita, ovvero `lowest-price`. L'`create-fleet` seguente crea una flotta utilizzando la strategia di allocazione. EC2 `lowest-price` Le tre specifiche di avvio, che sostituiscono il modello di avvio, hanno tipi di istanza diversi ma capacità ponderata e sottorete uguali. La capacità totale prevista è di 2 istanze e il modello di acquisto predefinito è `spot`. The EC2 Fleet lancia 2 istanze Spot utilizzando il tipo di istanza della specifica di lancio al prezzo più basso.

Quando crei un EC2 parco istanze, utilizza un JSON file per specificare le informazioni sulle istanze da avviare.

```
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"
  }
}

```

Output:

```

{
  "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}

```

- Per API i dettagli, consulta [CreateFleet AWS CLI Command Reference](#).

## create-flow-logs

Il seguente esempio di codice mostra come utilizzare `create-flow-logs`.

AWS CLI

Esempio 1: creare un registro di flusso

L'esempio seguente crea un log di flusso che acquisisce tutto il traffico rifiutato per l'interfaccia di rete specificata. I log di flusso vengono consegnati a un gruppo di log in CloudWatch Logs utilizzando le autorizzazioni nel ruolo specificato. IAM

```

aws ec2 create-flow-logs \
  --resource-type NetworkInterface \
  --resource-ids eni-11223344556677889 \
  --traffic-type REJECT \

```

```
--log-group-name my-flow-logs \  
--deliver-logs-permission-arn arn:aws:iam::123456789101:role/publishFlowLogs
```

Output:

```
{  
  "ClientToken": "so0eNA2uSHUN1HI0S2cJ305GuIX1CezaRdGtexample",  
  "FlowLogIds": [  
    "fl-12345678901234567"  
  ],  
  "Unsuccessful": []  
}
```

Per ulteriori informazioni, consulta [VPCFlow Logs](#) nella Amazon VPC User Guide.

Esempio 2: creare un log di flusso con un formato personalizzato

L'`create-flow-logs` seguente crea un log di flusso che acquisisce tutto il traffico per quanto specificato VPC e consegna i log di flusso a un bucket Amazon S3. Il parametro `--log-format` specifica un formato personalizzato per i record di log di flusso. Per eseguire questo comando su Windows, modifica le virgolette singole (') in virgolette doppie («).

```
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}'
```

Per ulteriori informazioni, consulta [VPCFlow Logs](#) nella Amazon VPC User Guide.

Esempio 3: creare un log di flusso con un intervallo di aggregazione massimo di un minuto

L'`create-flow-logs` seguente crea un log di flusso che acquisisce tutto il traffico per quanto specificato VPC e consegna i log di flusso a un bucket Amazon S3. Il `--max-aggregation-interval` parametro specifica un intervallo di aggregazione massimo di 60 secondi (1 minuto).

```
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
```

Per ulteriori informazioni, consulta [VPCFlow Logs](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [CreateFlowLogs AWS CLI Command Reference](#).

## create-fpga-image

Il seguente esempio di codice mostra come utilizzare `create-fpga-image`.

### AWS CLI

Per creare un'FPGA immagine Amazon

Questo esempio crea un AFI file dal tarball specificato nel bucket specificato.

Comando:

```
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
```

Output:

```
{  
  "FpgaImageId": "afi-0d123e123bfc85abc",  
  "FpgaImageGlobalId": "agfi-123cb27b5e84a0abc"  
}
```

- Per API i dettagli, vedere [CreateFpgaImage](#) in AWS CLI Command Reference.

## create-image

Il seguente esempio di codice mostra come utilizzare `create-image`.

## AWS CLI

Esempio 1: creare un file AMI da un'istanza EBS supportata da Amazon

L'`create-image` seguente crea un AMI dall'istanza specificata.

```
aws ec2 create-image \  
  --instance-id i-1234567890abcdef0 \  
  --name "My server" \  
  --description "An AMI for my server"
```

Output:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

Per ulteriori informazioni su come specificare una mappatura dei dispositivi a blocchi per il tuo AMI, consulta [Specifying a block device mapping per un dispositivo AMI nella Amazon User Guide](#).

### EC2

Esempio 2: creare un file AMI da un'istanza EBS supportata da Amazon senza riavviare

L'`create-image` seguente crea un parametro AMI e imposta il parametro `--no-reboot`, in modo che l'istanza non venga riavviata prima della creazione dell'immagine.

```
aws ec2 create-image \  
  --instance-id i-1234567890abcdef0 \  
  --name "My server" \  
  --no-reboot
```

Output:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

Per ulteriori informazioni su come specificare una mappatura dei dispositivi a blocchi per il tuo AMI, consulta [Specifying a block device mapping per un dispositivo AMI nella Amazon User Guide](#).

### EC2

### Esempio 3: etichettare un'AMI istantanea and al momento della creazione

L'`create-image` seguente crea un AMI e contrassegna le AMI istantanee con lo stesso tag `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}]"
```

Output:

```
{  
  "ImageId": "ami-abcdef01234567890"  
}
```

Per ulteriori informazioni sull'etichettatura delle risorse al momento della creazione, consulta [Aggiungere tag alla creazione di risorse](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [CreateImage AWS CLI Command Reference](#).

### **create-instance-connect-endpoint**

Il seguente esempio di codice mostra come utilizzare `create-instance-connect-endpoint`.

AWS CLI

Per creare un endpoint EC2 Instance Connect

L'`create-instance-connect-endpoint` seguente crea un endpoint EC2 Instance Connect nella sottorete specificata.

```
aws ec2 create-instance-connect-endpoint \  
  --region us-east-1 \  
  --subnet-id subnet-0123456789example
```

Output:

```
{
```

```

    "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"
  }

```

Per ulteriori informazioni, consulta [Create an EC2 Instance Connect Endpoint](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [CreateInstanceConnectEndpoint AWS CLI Command Reference](#).

## create-instance-event-window

Il seguente esempio di codice mostra come utilizzare `create-instance-event-window`.

### AWS CLI

Esempio 1: creare una finestra di eventi con un intervallo di tempo

L'esempio seguente crea una finestra di eventi con un intervallo di tempo. Non è possibile specificare anche il parametro `cron-expression`.

```

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
```

Output:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "TimeRanges": [  
      {  
        "StartWeekDay": "monday",  
        "StartHour": 2,  
        "EndWeekDay": "wednesday",  
        "EndHour": 8  
      }  
    ],  
    "Name": "myEventWindowName",  
    "State": "creating",  
    "Tags": [  
      {  
        "Key": "K1",  
        "Value": "V1"  
      }  
    ]  
  }  
}
```

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella sezione Eventi pianificati della Amazon EC2 User Guide.

Esempio 2: creare una finestra di eventi con un'espressione cron

L'`create-instance-event-window` seguente crea una finestra degli eventi con un'espressione cron. Non è possibile specificare anche il parametro `time-range`.

```
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
```

**Output:**

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "State": "creating",
    "Tags": [
      {
        "Key": "K1",
        "Value": "V1"
      }
    ]
  }
}
```

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella sezione Eventi pianificati della Amazon EC2 User Guide.

- Per API i dettagli, consulta Command [CreateInstanceEventWindow](#)Reference AWS CLI .

**create-instance-export-task**

Il seguente esempio di codice mostra come utilizzare `create-instance-export-task`.

**AWS CLI**

Per esportare un'istanza

Questo comando di esempio crea un'attività per esportare l'istanza `i-1234567890abcdef0` nel bucket Amazon S3 `myexportbucket`.

Comando:

```
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
```

**Output:**

```
{
```



```

"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"
}
}

```

- Per i AWS CLI dettagli, consulta Command [CreateInstanceExportTask](#)Reference. API

## create-internet-gateway

Il seguente esempio di codice mostra come utilizzare `create-internet-gateway`.

### AWS CLI

Per creare un gateway Internet

L'esempio seguente crea un gateway Internet con il tagName=my-igw.

```

aws ec2 create-internet-gateway \
  --tag-specifications ResourceType=internet-gateway,Tags=[{Key=Name,Value=my-igw}]

```

Output:

```

{
  "InternetGateway": {
    "Attachments": [],
    "InternetGatewayId": "igw-0d0fb496b3994d755",
    "OwnerId": "123456789012",
    "Tags": [

```

```

    {
      "Key": "Name",
      "Value": "my-igw"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Internet gateway](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [CreateInternetGateway AWS CLI Command Reference](#).

## create-ipam-pool

Il seguente esempio di codice mostra come utilizzare `create-ipam-pool`.

### AWS CLI

Per creare un IPAM pool

L'esempio seguente crea un 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"}]
```

Output:

```
{
  "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"
      }
    ]
  }
}
```

Per ulteriori informazioni, consulta la sezione [Piano per il provisioning degli indirizzi IP](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [CreateIpamPool AWS CLI Command Reference](#).

## create-ipam-resource-discovery

Il seguente esempio di codice mostra come utilizzare `create-ipam-resource-discovery`.

### AWS CLI

Per creare una scoperta di risorse

In questo esempio, sei un IPAM amministratore delegato che desidera creare e condividere un'individuazione delle risorse con l'IPAM amministratore di un'altra AWS organizzazione in modo che l'amministratore dell'altra organizzazione possa gestire e monitorare gli indirizzi IP delle risorse dell'organizzazione.

### Importante

Questo esempio include entrambe le `--operating-regions` opzioni `--region` e perché, sebbene siano facoltative, devono essere configurate in un modo particolare per integrare correttamente una scoperta di risorse con un.\* `--operating-regions` devono corrispondere alle regioni in cui sono presenti risorse che desideri IPAM scoprire IPAM. Se ci sono regioni in cui non desideri IPAM gestire gli indirizzi IP (ad esempio per motivi di conformità), non includerle. \* `--region` deve corrispondere alla regione di origine a IPAM cui desideri associarlo. È necessario creare l'individuazione delle risorse nella stessa regione in cui IPAM è stata creata. Ad esempio, se il file a IPAM cui ti stai associando è stato creato in `us-east-1`, `--region us-east-1` includi nella richiesta. Sia le opzioni che le opzioni sono predefinite `--region` sulla `--operating-regions` regione in cui stai eseguendo il comando se non le specifichi.

In questo esempio, le regioni operative con cui IPAM stiamo effettuando l'integrazione includono, e. `us-west-1 us-west-2 ap-south-1` Quando creiamo la scoperta delle risorse, vogliamo IPAM scoprire gli indirizzi IP delle risorse in `us-west-1` e `us-west-2` but not `ap-south-1`. Quindi lo includiamo solo `--operating-regions RegionName='us-west-1' RegionName='us-west-2'` nella richiesta.

L'`create-ipam-resource-discovery` esempio seguente crea una scoperta di IPAM risorse.

```
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
```

Output:

```
{
  "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"
      }
    ]
  }
}
```

Dopo aver creato una scoperta di risorse, potresti volerla condividere con un altro amministratore IPAM delegato, operazione che puoi fare. [create-resource-share](#) Per ulteriori informazioni, consulta [Integrazione IPAM con account esterni alla tua organizzazione](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [CreateIpamResourceDiscovery AWS CLI](#) Command Reference.

## create-ipam-scope

Il seguente esempio di codice mostra come utilizzare `create-ipam-scope`.

### AWS CLI

Per creare un IPAM ambito

L'create-ipam-scope seguente crea un IPAM ambito.

(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"}]
```

Output:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Creare ambiti aggiuntivi](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [CreateIpamScope AWS CLI Command Reference](#).

## create-ipam

Il seguente esempio di codice mostra come utilizzare `create-ipam`.

### AWS CLI

Per creare un IPAM

L'esempio seguente crea un 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}]
```

Output:

```
{  
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Create an IPAM](#) in the Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [CreateIpam AWS CLI Command Reference](#).

## create-key-pair

Il seguente esempio di codice mostra come utilizzare `create-key-pair`.

### AWS CLI

Per creare una coppia di chiavi

Nell'esempio seguente viene creata una coppia di chiavi denominata `MyKeyPair`.

Comando:

```
aws ec2 create-key-pair --key-name MyKeyPair
```

L'output è una ASCII versione della chiave privata e dell'impronta digitale della chiave. È necessario salvare la chiave in un file.

Per ulteriori informazioni, consulta [Utilizzo delle coppie di chiavi](#) nella Guida per l'utente dell'Interfaccia a riga di comando AWS .

- Per API i dettagli, vedere [CreateKeyPair](#) in AWS CLI Command Reference.



## create-launch-template-version

Il seguente esempio di codice mostra come utilizzare `create-launch-template-version`.

### AWS CLI

Per creare una versione del modello di lancio

Questo esempio crea una nuova versione del modello di lancio basata sulla versione 1 del modello di lancio e specifica un AMI ID diverso.

Comando:

```
aws ec2 create-launch-template-version --launch-template-id lt-0abcd290751193123
--version-description WebVersion2 --source-version 1 --launch-template-data
'{"ImageId": "ami-c998b6b2"}'
```

Output:

```
{
  "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"
  }
}
```

```
}
}
```

- Per API i dettagli, consulta [CreateLaunchTemplateVersion AWS CLI Command Reference](#).

## create-launch-template

Il seguente esempio di codice mostra come utilizzare `create-launch-template`.

### AWS CLI

Esempio 1: per creare un modello di avvio

L'esempio seguente crea un modello di avvio che specifica la sottorete in cui avviare l'istanza, assegna un indirizzo IP pubblico e un IPv6 indirizzo all'istanza e crea un tag per l'istanza.

```
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"}]}]}'
```

Output:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-01238c059e3466abc",
    "LaunchTemplateName": "TemplateForWebServer",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2019-01-27T09:13:24.000Z"
  }
}
```

Per maggiori informazioni, consulta [Avvio di istanze da un modello di avvio](#) nella Guida per l'utente di Amazon Elastic Compute Cloud. Per informazioni sulla quotazione dei parametri JSON `-formatted`, vedete [Quoting Strings](#) nella [Command Line Interface User Guide](#).AWS

Esempio 2: creare un modello di lancio per Amazon EC2 Auto Scaling

L'`create-launch-template` seguente crea un modello di lancio con più tag e una mappatura dei dispositivi a blocchi per specificare un EBS volume aggiuntivo all'avvio di un'istanza. Specificate un valore `Groups` corrispondente ai gruppi di sicurezza in VPC cui il gruppo Auto Scaling lancerà le istanze. Specificate le sottoreti VPC and come proprietà del gruppo Auto Scaling.

```
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
```

Output:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 1,
    "LaunchTemplateId": "lt-0123c79c33a54e0abc",
    "LaunchTemplateName": "TemplateForAutoScaling",
    "DefaultVersionNumber": 1,
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2019-04-30T18:16:06.000Z"
  }
}
```

Per ulteriori informazioni, consulta [Creating a Launch Template for an Auto Scaling Group](#) nella [Amazon Auto EC2 Scaling User Guide](#). Per informazioni su come quotare i parametri JSON - formatted, consulta [Quoting Strings](#) nella [Command Line Interface User Guide](#).AWS

Esempio 3: creare un modello di avvio che specifichi la crittografia dei volumi EBS

L'`create-launch-template` seguente crea un modello di avvio che include EBS volumi crittografati creati da un'istanza non crittografata. Inoltre, vengono applicati i tag ai volumi durante la creazione. Se la crittografia predefinita è disabilitata, è necessario

specificare l'opzione "Encrypted" come mostrato nel seguente esempio. Se si utilizza l'"KmsKeyId" opzione per specificare un servizio gestito dal cliente CMK, è necessario specificare l'"Encrypted" opzione anche se la crittografia è abilitata per impostazione predefinita.

```
aws ec2 create-launch-template \  
  --launch-template-name TemplateForEncryption \  
  --launch-template-data file://config.json
```

Contenuto di 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"  
        }  
      ]  
    }  
  ]  
}
```

Output:

```
{
```

```

    "LaunchTemplate": {
      "LatestVersionNumber": 1,
      "LaunchTemplateId": "lt-0d5bd51bcf8530abc",
      "LaunchTemplateName": "TemplateForEncryption",
      "DefaultVersionNumber": 1,
      "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
      "CreateTime": "2020-01-07T19:08:36.000Z"
    }
  }
}

```

Per ulteriori informazioni, consulta [Restoring an Amazon EBS Volume from a Snapshot and Encryption by Default](#) nella [Amazon Elastic Compute Cloud User Guide](#).

- Per API i dettagli, consulta [Command `CreateLaunchTemplate` Reference AWS CLI](#).

## create-local-gateway-route-table-virtual-interface-group-association

Il seguente esempio di codice mostra come utilizzare `create-local-gateway-route-table-virtual-interface-group-association`.

### AWS CLI

Per associare una tabella di routing del gateway locale a un gruppo di interfacce virtuali (VIFs)

L'esempio seguente crea un'associazione tra la tabella di routing del gateway locale e il VIF gruppo specificati.

```

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

```

### Output:

```

{
  "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": []
  }
}

```

Per ulteriori informazioni, consulta le [associazioni di VIF gruppo](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, consulta [CreateLocalGatewayRouteTableVirtualInterfaceGroupAssociation](#) [AWS CLI Command Reference](#).

## create-local-gateway-route-table-vpc-association

Il seguente esempio di codice mostra come utilizzare `create-local-gateway-route-table-vpc-association`.

### AWS CLI

Associare a VPC a una tabella di percorsi

L'`create-local-gateway-route-table-vpc-association` esempio seguente associa la tabella di route del gateway locale specificata VPC alla tabella di routing del gateway locale specificata.

```

aws ec2 create-local-gateway-route-table-vpc-association \
  --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
  --vpc-id vpc-07ef66ac71EXAMPLE

```

Output:

```

{
  "LocalGatewayRouteTableVpcAssociation": {
    "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0ee765bcc8EXAMPLE",
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
    "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
    "VpcId": "vpc-07ef66ac71EXAMPLE",
    "State": "associated"
  }
}

```

```
}
```

- Per API i dettagli, vedere [CreateLocalGatewayRouteTableVpcAssociation](#) in AWS CLI Command Reference.

## create-local-gateway-route-table

Il seguente esempio di codice mostra come utilizzare `create-local-gateway-route-table`.

### AWS CLI

Per creare una tabella di routing del gateway locale

L'esempio seguente crea una tabella di routing del gateway locale con la modalità di VPC routing diretto.

```
aws ec2 create-local-gateway-route-table \
  --local-gateway-id lgw-1a2b3c4d5e6f7g8h9 \
  --mode direct-vpc-routing
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Local gateway route tables](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, vedere [CreateLocalGatewayRouteTable](#) in AWS CLI Command Reference.

## create-local-gateway-route

Il seguente esempio di codice mostra come utilizzare `create-local-gateway-route`.

### AWS CLI

Per creare una route statica per una tabella di routing del gateway locale

L'esempio seguente crea la route specificata nella tabella di routing del gateway locale specificata.

```
aws ec2 create-local-gateway-route \
  --destination-cidr-block 0.0.0.0/0 \
  --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE
```

Output:

```
{
  "Route": {
    "DestinationCidrBlock": "0.0.0.0/0",
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
    "Type": "static",
    "State": "deleted",
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE"
  }
}
```

- Per API i dettagli, vedere [CreateLocalGatewayRoute](#) in AWS CLI Command Reference.

## create-managed-prefix-list

Il seguente esempio di codice mostra come utilizzare `create-managed-prefix-list`.

### AWS CLI

Per creare un elenco di prefissi

L'esempio seguente crea un elenco di IPv4 prefissi con un massimo di 10 voci e ne crea 2 nell'elenco dei prefissi.

```
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

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Managed Prefix Lists](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [CreateManagedPrefixList AWS CLI Command Reference](#).

## create-nat-gateway

Il seguente esempio di codice mostra come utilizzare `create-nat-gateway`.

### AWS CLI

Esempio 1: creare un NAT gateway pubblico

L'esempio seguente crea un NAT gateway pubblico nella sottorete specificata e associa l'indirizzo IP elastico all'ID di allocazione specificato. Quando si crea un NAT gateway pubblico, è necessario associare un indirizzo IP elastico.

```

aws ec2 create-nat-gateway \
  --subnet-id subnet-0250c25a1fEXAMPLE \
  --allocation-id eipalloc-09ad461b0dEXAMPLE

```

**Output:**

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta i [NATgateway](#) nella Amazon VPC User Guide.

**Esempio 2: creare un gateway privato NAT**

L'create-nat-gatewayesempio seguente crea un NAT gateway privato nella sottorete specificata. Un NAT gateway privato non ha un indirizzo IP elastico associato.

```
aws ec2 create-nat-gateway \
  --subnet-id subnet-0250c25a1fEXAMPLE \
  --connectivity-type private
```

**Output:**

```
{
  "NatGateway": {
    "CreateTime": "2021-12-01T22:26:00.000Z",
    "NatGatewayAddresses": [
      {}
    ],
    "NatGatewayId": "nat-011b568379EXAMPLE",
    "State": "pending",
    "SubnetId": "subnet-0250c25a1fEXAMPLE",
    "VpcId": "vpc-0a60eb65b4EXAMPLE",
    "ConnectivityType": "private"
  }
}
```

```
}
```

Per ulteriori informazioni, consulta i [NATgateway](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [CreateNatGateway AWS CLI](#) Command Reference.

## **create-network-acl-entry**

Il seguente esempio di codice mostra come utilizzare `create-network-acl-entry`.

### AWS CLI

Per creare una ACL voce di rete

Questo esempio crea una voce per la rete specificata ACL. La regola consente il traffico in ingresso da qualsiasi IPv4 indirizzo (0.0.0.0/0) sulla UDP porta 53 (DNS) in qualsiasi sottorete associata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
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
```

Questo esempio crea una regola per la rete specificata ACL che consente il traffico in ingresso da qualsiasi IPv6 indirizzo (:: /0) sulla porta 80 (). TCP HTTP

Comando:

```
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
```

- Per API i dettagli, vedere [CreateNetworkAcEntry](#) in AWS CLI Command Reference.

## **create-network-acl**

Il seguente esempio di codice mostra come utilizzare `create-network-acl`.

### AWS CLI

Per creare una rete ACL

Questo esempio crea una rete ACL per quanto specificato VPC.

Comando:

```
aws ec2 create-network-acl --vpc-id vpc-a01106c2
```

Output:

```
{
  "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
  }
}
```

- Per API i dettagli, vedere [CreateNetworkAcl](#) in AWS CLI Command Reference.

## **create-network-insights-access-scope**

Il seguente esempio di codice mostra come utilizzare `create-network-insights-access-scope`.

## AWS CLI

Per creare un ambito di accesso alla rete

L'create-network-insights-access-scopeesempio seguente crea un Network Access Scope.

```
aws ec2 create-network-insights-access-scope \  
  --cli-input-json file://access-scope-file.json
```

Contenuto di access-scope-file.json.

```
{  
  "MatchPaths": [  
    {  
      "Source": {  
        "ResourceStatement": {  
          "Resources": [  
            "vpc-abcd12e3"  
          ]  
        }  
      }  
    },  
  ],  
  "ExcludePaths": [  
    {  
      "Source": {  
        "ResourceStatement": {  
          "ResourceTypes": [  
            "AWS::EC2::InternetGateway"  
          ]  
        }  
      }  
    }  
  ]  
}
```

Output:

```
{  
  "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"
            ]
          }
        }
      }
    ]
  }
}

```

Per ulteriori informazioni, vedere [Guida introduttiva all'uso di Network Access Analyzer utilizzando la Guida AWS CLI di Network Access Analyzer](#).

- Per API i dettagli, vedere [CreateNetworkInsightsAccessScope](#) in AWS CLI Command Reference.

## create-network-insights-path

Il seguente esempio di codice mostra come utilizzare `create-network-insights-path`.

## AWS CLI

Per creare un percorso

L'`create-network-insights-path` seguente crea un percorso. L'origine è il gateway Internet specificato e la destinazione è l'EC2istanza specificata. Per determinare se la destinazione è raggiungibile utilizzando il protocollo e la porta specificati, analizza il percorso utilizzando il `start-network-insights-analysis` comando.

```
aws ec2 create-network-insights-path \  
  --source igw-0797cccdc9d73b0e5 \  
  --destination i-0495d385ad28331c7 \  
  --destination-port 22 \  
  --protocol TCP
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta la [Guida introduttiva al AWS CLI Reachability Analyzer Guide](#).

- Per API i dettagli, vedere [CreateNetworkInsightsPath](#) in AWS CLI Command Reference.

## **create-network-interface-permission**

Il seguente esempio di codice mostra come utilizzare `create-network-interface-permission`.

## AWS CLI

Per creare un'autorizzazione all'interfaccia di rete

Questo esempio concede all'account l'autorizzazione 123456789012 a collegare l'interfaccia di rete eni-1a2b3c4d a un'istanza.

Comando:

```
aws ec2 create-network-interface-permission --network-interface-id eni-1a2b3c4d --  
aws-account-id 123456789012 --permission INSTANCE-ATTACH
```

Output:

```
{  
  "InterfacePermission": {  
    "PermissionState": {  
      "State": "GRANTED"  
    },  
    "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",  
    "NetworkInterfaceId": "eni-1a2b3c4d",  
    "Permission": "INSTANCE-ATTACH",  
    "AwsAccountId": "123456789012"  
  }  
}
```

- Per API i dettagli, vedere [CreateNetworkInterfacePermission](#) in AWS CLI Command Reference.

## create-network-interface

Il seguente esempio di codice mostra come utilizzare `create-network-interface`.

AWS CLI

Esempio 1: specificare un IPv4 indirizzo per un'interfaccia di rete

L'`create-network-interface` esempio seguente crea un'interfaccia di rete per la sottorete specificata con l'IPv4 indirizzo primario specificato.

```
aws ec2 create-network-interface \  
  --subnet-id subnet-00a24d0d67acf6333 \  
  --description "my network interface" \  
  --groups sg-09dfba7ed20cda78b \  
  --private-ip-address 10.0.8.17
```

Output:



```
{
  "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"
  }
}
```

Esempio 2: creare un'interfaccia di rete con un IPv4 indirizzo e un IPv6 indirizzo

L'create-network-interfaceesempio seguente crea un'interfaccia di rete per la sottorete specificata con un IPv4 indirizzo e un IPv6 indirizzo selezionati da AmazonEC2.

```
aws ec2 create-network-interface \
  --subnet-id subnet-00a24d0d67acf6333 \
  --description "my dual stack network interface" \
  --ipv6-address-count 1 \
```

```
--groups sg-09dfba7ed20cda78b
```

## Output:

```
{
  "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"
  }
}
```

### Esempio 3: creare un'interfaccia di rete con opzioni di configurazione per il tracciamento delle connessioni

L'create-network-interfaceesempio seguente crea un'interfaccia di rete e configura i timeout di tracciamento delle connessioni inattive.

```
aws ec2 create-network-interface \  
  --subnet-id subnet-00a24d0d67acf6333 \  
  --groups sg-02e57dbcf0331c1b \  
  --connection-tracking-specification TcpEstablishedTimeout=86400,UdpTimeout=60
```

Output:

```
{  
  "NetworkInterface": {  
    "AvailabilityZone": "us-west-2a",  
    "ConnectionTrackingConfiguration": {  
      "TcpEstablishedTimeout": 86400,  
      "UdpTimeout": 60  
    },  
    "Description": "",  
    "Groups": [  
      {  
        "GroupName": "my-security-group",  
        "GroupId": "sg-02e57dbcf0331c1b"  
      }  
    ],  
    "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-02723a0feeeb9d57b"
  }
}

```

#### Esempio 4: creare un Elastic Fabric Adapter

L'create-network-interfaceesempio seguente crea unEFA.

```

aws ec2 create-network-interface \
  --interface-type efa \
  --subnet-id subnet-00a24d0d67acf6333 \
  --description "my efa" \
  --groups sg-02e57dbcfe0331c1b

```

Output:

```

{
  "NetworkInterface": {
    "AvailabilityZone": "us-west-2a",
    "Description": "my efa",
    "Groups": [
      {
        "GroupName": "my-efa-sg",
        "GroupId": "sg-02e57dbcfe0331c1b"
      }
    ],
    "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Interfacce di rete elastiche](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [CreateNetworkInterface AWS CLI Command Reference](#).

## create-placement-group

Il seguente esempio di codice mostra come utilizzare `create-placement-group`.

### AWS CLI

Per creare un gruppo di collocamento

Questo comando di esempio crea un gruppo di posizionamento con il nome specificato.

Comando:

```
aws ec2 create-placement-group --group-name my-cluster --strategy cluster
```

Per creare un gruppo di posizionamento delle partizioni

Questo comando di esempio crea un gruppo di posizionamento delle partizioni denominato `HDFS-Group-A` con cinque partizioni.

Comando:

```
aws ec2 create-placement-group --group-name HDFS-Group-A --strategy partition --  
partition-count 5
```

- Per API i dettagli, vedere [CreatePlacementGroup](#) in AWS CLI Command Reference.

## create-replace-root-volume-task

Il seguente esempio di codice mostra come utilizzare `create-replace-root-volume-task`.

### AWS CLI

Esempio 1: per ripristinare un volume root allo stato di avvio iniziale

L'esempio seguente ripristina il volume root dell'istanza `i-0123456789abcdefa` allo stato di avvio iniziale.

```
aws ec2 create-replace-root-volume-task \  
  --instance-id i-0123456789abcdefa
```

Output:

```
{  
  "ReplaceRootVolumeTask":  
  {  
    "InstanceId": "i-0123456789abcdefa",  
    "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",  
    "TaskState": "pending",  
    "StartTime": "2022-03-14T15:06:38Z",  
    "Tags": []  
  }  
}
```

Per ulteriori informazioni, consulta [Sostituire un volume root](#) nella Guida per l'utente di Amazon Elastic Compute Cloud.

Esempio 2: ripristinare un volume root in uno snapshot specifico

L'esempio seguente ripristina il volume principale dell'istanza `i-0123456789abcdefa` nello snapshot `snap-0abcdef1234567890`.

```
aws ec2 create-replace-root-volume-task \  
  --instance-id i-0123456789abcdefa \  
  --snapshot-id snap-0abcdef1234567890
```

Output:

```
{
```

```
"ReplaceRootVolumeTask":
{
  "InstanceId": "i-0123456789abcdefa",
  "ReplaceRootVolumeTaskId": "replacevol-0555566667777abcd",
  "TaskState": "pending",
  "StartTime": "2022-03-14T15:16:28Z",
  "Tags": []
}
```

Per ulteriori informazioni, consulta [Sostituire un volume root](#) nella Guida per l'utente di Amazon Elastic Compute Cloud.

- Per API i dettagli, consulta [CreateReplaceRootVolumeTask AWS CLI Command Reference](#).

## create-reserved-instances-listing

Il seguente esempio di codice mostra come utilizzare `create-reserved-instances-listing`.

### AWS CLI

Per elencare un'istanza riservata nel Marketplace delle istanze riservate

L'esempio seguente crea un elenco per l'istanza riservata specificata nel Reserved Instance Marketplace.

```
aws ec2 create-reserved-instances-listing \
  --reserved-instances-id 5ec28771-05ff-4b9b-aa31-9e57dexample \
  --instance-count 3 \
  --price-schedules CurrencyCode=USD,Price=25.50 \
  --client-token 550e8400-e29b-41d4-a716-446655440000
```

- Per API i dettagli, vedere [CreateReservedInstancesListing](#) in AWS CLI Command Reference.

## create-restore-image-task

Il seguente esempio di codice mostra come utilizzare `create-restore-image-task`.

### AWS CLI

Per ripristinare un file AMI da un bucket S3

L'create-restore-image-task esempio seguente ripristina un oggetto AMI da un bucket S3. Utilizzate i S3objectKey `` and ``Bucket valori dell'describe-store-image-task output, specificate la chiave oggetto del AMI e il nome del bucket S3 in cui AMI è stato copiato e specificate il nome del file ripristinato. AMI Il nome deve essere univoco AMIs nella regione per questo account. Il ripristinato AMI riceverà un nuovo AMI ID.

```
aws ec2 create-restore-image-task \  
  --object-key ami-1234567890abcdef0.bin \  
  --bucket my-ami-bucket \  
  --name "New AMI Name"
```

Output:

```
{  
  "ImageId": "ami-0eab20fe36f83e1a8"  
}
```

Per ulteriori informazioni sull'archiviazione e il ripristino di un dispositivo AMI utilizzando S3, consulta Archiviare e ripristinare un AMI utilizzo di S3 <<https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/ami-store-restore.html>> nella Amazon User Guide. EC2

- Per API [CreateRestoreImageTask](#) AWS CLI dettagli, consulta Command Reference.

## create-route-table

Il seguente esempio di codice mostra come utilizzare create-route-table.

AWS CLI

Per creare una tabella di routing

Questo esempio crea una tabella di routing per il percorso specificato VPC.

Comando:

```
aws ec2 create-route-table --vpc-id vpc-a01106c2
```

Output:

```
{
```



```

"RouteTable": {
  "Associations": [],
  "RouteTableId": "rtb-22574640",
  "VpcId": "vpc-a01106c2",
  "PropagatingVgws": [],
  "Tags": [],
  "Routes": [
    {
      "GatewayId": "local",
      "DestinationCidrBlock": "10.0.0.0/16",
      "State": "active"
    }
  ]
}
}

```

- Per API i dettagli, vedere [CreateRouteTable](#) in AWS CLI Command Reference.

## create-route

Il seguente esempio di codice mostra come utilizzare `create-route`.

### AWS CLI

Per creare un percorso

Questo esempio crea un percorso per la tabella di rotte specificata. La route corrisponde a tutto IPv4 il traffico (`0.0.0.0/0`) e lo indirizza verso il gateway Internet specificato. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 create-route --route-table-id rtb-22574640 --destination-cidr-block 0.0.0.0/0 --gateway-id igw-c0a643a9
```

Questo comando di esempio crea una route nella tabella delle rotte `rtb-g8ff4ea2`. La route corrisponde al traffico per il IPv4 CIDR blocco `10.0.0.0/16` e lo indirizza alla connessione peering, `pcx-111aaa22`. VPC Questo percorso consente di indirizzare il traffico verso il peer nella connessione peering. VPC VPC Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 create-route --route-table-id rtb-g8ff4ea2 --destination-cidr-block 10.0.0.0/16 --vpc-peering-connection-id pcx-1a2b3c4d
```

Questo esempio crea una route nella tabella di route specificata che corrisponde a tutto il IPv6 traffico (: : /0) e la indirizza verso il gateway Internet di sola uscita specificato.

Comando:

```
aws ec2 create-route --route-table-id rtb-dce620b8 --destination-ipv6-cidr-block ::/0 --egress-only-internet-gateway-id eigw-01eadbd45ecd7943f
```

- Per API i dettagli, vedere [CreateRoute](#) in AWS CLI Command Reference.

## create-security-group

Il seguente esempio di codice mostra come utilizzare `create-security-group`.

AWS CLI

Per creare un gruppo di sicurezza per EC2 -Classic

Nell'esempio seguente viene creato un gruppo di sicurezza denominato `MySecurityGroup`.

Comando:

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My security group"
```

Output:

```
{
  "GroupId": "sg-903004f8"
}
```

Per creare un gruppo di sicurezza per EC2 - VPC

Questo esempio crea un gruppo di sicurezza denominato `MySecurityGroup` base a quanto specificato VPC.

Comando:

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My security group" --vpc-id vpc-1a2b3c4d
```

Output:

```
{
  "GroupId": "sg-903004f8"
}
```

Per ulteriori informazioni, consulta [Utilizzo dei gruppi di sicurezza](#) nella Guida per l'utente dell'Interfaccia a riga di comando AWS .

- Per API i dettagli, vedere [CreateSecurityGroup](#) in AWS CLI Command Reference.

## create-snapshot

Il seguente esempio di codice mostra come utilizzare `create-snapshot`.

AWS CLI

Per creare un'istantanea

Questo comando di esempio crea un'istantanea del volume con un ID di volume `vol-1234567890abcdef0` e una breve descrizione per identificare l'istantanea.

Comando:

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description "This is my root volume snapshot"
```

Output:

```
{
  "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"
}
```

Per creare un'istantanea con tag

Questo comando di esempio crea un'istantanea e applica due tag: `purpose=prod` e `costcenter=123`.

Comando:

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description 'Prod backup' --tag-specifications 'ResourceType=snapshot,Tags=[{Key=purpose,Value=prod},{Key=costcenter,Value=123}]'
```

Output:

```
{
  "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"
}
```

- API Per i [CreateSnapshot AWS CLI](#) dettagli, vedere in Command Reference.

## create-snapshots

Il seguente esempio di codice mostra come utilizzare `create-snapshots`.

### AWS CLI

Esempio 1: per creare un'istantanea multivolume

L'esempio seguente crea istantanee di tutti i volumi collegati all'istanza specificata.

```
aws ec2 create-snapshots \  
  --instance-specification InstanceId=i-1234567890abcdef0 \  
  --description "This is snapshot of a volume from my-instance"
```

Output:

```
{  
  "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"  
    },  
  ],  
}
```

```

    ...
  ]
}

```

Esempio 2: creare un'istantanea multivolume con tag dal volume di origine

L'`create-snapshots` seguente crea istantanee di tutti i volumi collegati all'istanza specificata e copia i tag da ciascun volume nell'istantanea corrispondente.

```

aws ec2 create-snapshots \
  --instance-specification InstanceId=i-1234567890abcdef0 \
  --copy-tags-from-source volume \
  --description "This is snapshot of a volume from my-instance"

```

Output:

```

{
  "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"
    }
    ...
  ]
}

```

Esempio 3: creare un'istantanea multivolume che non includa il volume principale

L'`create-snapshots` seguente crea un'istantanea di tutti i volumi collegati all'istanza specificata ad eccezione del volume principale.

```
aws ec2 create-snapshots \  
  --instance-specification InstanceId=i-1234567890abcdef0,ExcludeBootVolume=true
```

Per un esempio di output, vedete l'esempio 1.

Esempio 4: per creare un'istantanea multivolume e aggiungere tag

L'`create-snapshots` seguente crea istantanee di tutti i volumi collegati all'istanza specificata e aggiunge due tag a ciascuna istantanea.

```
aws ec2 create-snapshots \  
  --instance-specification InstanceId=i-1234567890abcdef0 \  
  --tag-specifications 'ResourceType=snapshot,Tags=[{Key=Name,Value=backup},  
{Key=costcenter,Value=123}]'
```

Vedi l'esempio 1 per un output di esempio.

- Per API i dettagli, vedere [CreateSnapshots](#) in AWS CLI Command Reference.

## **create-spot-datafeed-subscription**

Il seguente esempio di codice mostra come utilizzare `create-spot-datafeed-subscription`.

### AWS CLI

Per creare un data feed di istanze Spot

L'`create-spot-datafeed-subscription` seguente crea un feed di dati di istanze Spot.

```
aws ec2 create-spot-datafeed-subscription \  
  --bucket my-bucket \  
  --prefix spot-data-feed
```

Output:

```
{  
  "SpotDatafeedSubscription": {
```

```
"Bucket": "my-bucket",
"OwnerId": "123456789012",
"Prefix": "spot-data-feed",
"State": "Active"
}
}
```

Il data feed è archiviato nel bucket Amazon S3 che hai specificato. I nomi di file per questo feed di dati hanno il seguente formato.

```
my-bucket.s3.amazonaws.com/spot-data-feed/123456789012.YYYY-MM-DD-HH.n.abcd1234.gz
```

Per ulteriori informazioni, consulta il [data feed di istanze Spot](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per API i dettagli, consulta AWS CLI Command [CreateSpotDatafeedSubscription](#) Reference.

## create-store-image-task

Il seguente esempio di codice mostra come utilizzare `create-store-image-task`.

### AWS CLI

Per archiviare un file AMI in un bucket S3

L'esempio seguente memorizza un AMI in un bucket S3. Specificare l'ID AMI e il nome del bucket S3 in cui archiviare il. AMI

```
aws ec2 create-store-image-task \
  --image-id ami-1234567890abcdef0 \
  --bucket my-ami-bucket
```

Output:

```
{
  "ObjectKey": "ami-1234567890abcdef0.bin"
}
```

Per ulteriori informazioni, consulta [Archiviare e ripristinare e AMI utilizzare S3](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [CreateStoreImageTask AWS CLI](#) Command Reference.



## create-subnet-cidr-reservation

Il seguente esempio di codice mostra come utilizzare `create-subnet-cidr-reservation`.

### AWS CLI

Per creare una prenotazione di sottorete CIDR

L'esempio seguente crea una CIDR prenotazione di sottorete per la sottorete e l'intervallo specificati. CIDR

```
aws ec2 create-subnet-cidr-reservation \
  --subnet-id subnet-03c51e2eEXAMPLE \
  --reservation-type prefix \
  --cidr 10.1.0.20/26
```

Output:

```
{
  "SubnetCidrReservation": {
    "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
    "SubnetId": "subnet-03c51e2e6cEXAMPLE",
    "Cidr": "10.1.0.16/28",
    "ReservationType": "prefix",
    "OwnerId": "123456789012"
  }
}
```

Per ulteriori informazioni, consulta [Subnet CIDR reservations](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [CreateSubnetCidrReservation AWS CLI Command Reference](#).

## create-subnet

Il seguente esempio di codice mostra come utilizzare `create-subnet`.

### AWS CLI

Esempio 1: creare una sottorete con un solo IPv4 CIDR blocco

L'esempio seguente crea una sottorete nel blocco specificato VPC con il blocco specificato IPv4CIDR.

```
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}]
```

Output:

```
{
  "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"
  }
}
```

Esempio 2: creare una sottorete con entrambi i blocchi IPv4 IPv6 CIDR

L'create-subnetesempio seguente crea una sottorete nel campo specificato VPC con i blocchi IPv4 e IPv6 CIDR specificati.

```
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}]
```

Output:

```
{
  "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"
  }
}
```

Esempio 3: creare una sottorete con un solo blocco IPv6 CIDR

L'create-subnetesempio seguente crea una sottorete nel blocco specificato VPC con il blocco specificato IPv6CIDR.

```
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}]
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [VPCse sottoreti](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateSubnetReference](#).

## create-tags

Il seguente esempio di codice mostra come utilizzare `create-tags`.

### AWS CLI

Esempio 1: per aggiungere un tag a una risorsa

L'`create-tags`esempio seguente aggiunge il tag `Stack=production` all'immagine specificata o sovrascrive un tag esistente per la chiave AMI where the tag. `Stack`

```
aws ec2 create-tags \  
  --resources ami-1234567890abcdef0 \  
  --tags Key=Stack,Value=production
```

Per ulteriori informazioni, consulta [Questo è il titolo dell'argomento](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

Esempio 2: per aggiungere tag a più risorse

L'`create-tags`esempio seguente aggiunge (o sovrascrive) due tag per un'istanza AMI e un'istanza. Uno dei tag ha una chiave (`webserver`), ma nessun valore (il valore è impostato su una stringa vuota). L'altro tag ha una chiave (`stack`) e un valore (`Production`).

```
aws ec2 create-tags \  
  --resources ami-1a2b3c4d i-1234567890abcdef0 \  
  --tags Key=webserver,Value= Key=stack,Value=Production
```

Per ulteriori informazioni, consulta [Questo è il titolo dell'argomento](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

Esempio 3: per aggiungere tag contenenti caratteri speciali

Nell'esempio di `create-tags` seguente vien aggiunto il tag `[Group]=test` a un'istanza. Le parentesi quadre (`[` e `]`) sono caratteri speciali per i quali occorre eseguire l'escape. Negli esempi seguenti viene utilizzato anche il carattere di continuazione della riga adeguato per ogni ambiente.

Se si utilizza Windows, racchiudere l'elemento con caratteri speciali tra virgolette doppie (`"`), quindi anteporre ad ogni carattere virgolette doppie una barra rovesciata (`\`) come segue:

```
aws ec2 create-tags ^
  --resources i-1234567890abcdef0 ^
  --tags Key=\"[Group]\",Value=test
```

Se utilizzate Windows PowerShell, racchiudete l'elemento con caratteri speciali tra virgolette doppie («), fate precedere ogni virgoletta doppia da una barra rovesciata (\), quindi racchiudete l'intera struttura della chiave e del valore tra virgolette singole (') come segue:

```
aws ec2 create-tags `
  --resources i-1234567890abcdef0 `
  --tags 'Key=\"[Group]\",Value=test'
```

Se si utilizza Linux o OS X, racchiudere l'elemento con caratteri speciali con virgolette doppie ("), quindi racchiudere l'intera struttura chiave e valore tra virgolette singole ('), come segue:

```
aws ec2 create-tags \
  --resources i-1234567890abcdef0 \
  --tags 'Key="[Group]",Value=test'
```

Per ulteriori informazioni, consulta [Questo è il titolo dell'argomento](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

- Per API i dettagli, consulta AWS CLI Command [CreateTagsReference](#).

## create-traffic-mirror-filter-rule

Il seguente esempio di codice mostra come utilizzare `create-traffic-mirror-filter-rule`.

### AWS CLI

Per creare una regola di filtro per il traffico in entrata TCP

L'`create-traffic-mirror-filter-rule` esempio seguente crea una regola che è possibile utilizzare per rispecchiare tutto il traffico in entrata TCP. Prima di eseguire questo comando, utilizzalo `create-traffic-mirror-filter` per creare il filtro Traffic Mirror.

```
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

```

Output:

```

{
  "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"
}

```

Per ulteriori informazioni, consulta [Creare un filtro Traffic Mirror](#) nella AWS Traffic Mirroring Guide.

- Per API i dettagli, consulta [CreateTrafficMirrorFilterRule AWS CLI Command Reference](#).

## create-traffic-mirror-filter

Il seguente esempio di codice mostra come utilizzare `create-traffic-mirror-filter`.

### AWS CLI

Per creare un filtro Traffic Mirror

L'`create-traffic-mirror-filter` esempio seguente crea un filtro Traffic Mirror. Dopo aver creato il filtro, `create-traffic-mirror-filter-rule` utilizzalo per aggiungere regole al filtro.

```

aws ec2 create-traffic-mirror-filter \
  --description "TCP Filter"

```

**Output:**

```
{
  "ClientToken": "28908518-100b-4987-8233-8c744EXAMPLE",
  "TrafficMirrorFilter": {
    "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
    "Description": "TCP Filter",
    "EgressFilterRules": [],
    "IngressFilterRules": [],
    "Tags": [],
    "NetworkServices": []
  }
}
```

Per ulteriori informazioni, consulta [Creare un filtro Traffic Mirror](#) nella AWS Traffic Mirroring Guide.

- Per API i dettagli, consulta [CreateTrafficMirrorFilter AWS CLI Command Reference](#).

**create-traffic-mirror-session**

Il seguente esempio di codice mostra come utilizzare `create-traffic-mirror-session`.

**AWS CLI**

Per creare una sessione di Traffic Mirror

Il `create-traffic-mirror-session` comando seguente crea una sessione di traffic mirror per l'origine e la destinazione specificate per 25 byte del pacchetto.

```
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
```

**Output:**

```
{
  "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"
}

```

Per ulteriori informazioni, consulta [Creare una sessione Traffic Mirror](#) nella Traffic Mirroring AWS Guide.

- Per API i dettagli, consulta [CreateTrafficMirrorSession AWS CLI Command Reference](#).

## create-traffic-mirror-target

Il seguente esempio di codice mostra come utilizzare `create-traffic-mirror-target`.

### AWS CLI

Per creare un target Network Load Balancer Traffic Mirror

L'esempio seguente crea un target Network Load Balancer Traffic Mirror.

```

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

```

Output:

```

{
  "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"
}

```

Per creare un target Traffic Mirror di rete

L'`create-traffic-mirror-target` seguente crea un target Traffic Mirror per l'interfaccia di rete.

```
aws ec2 create-traffic-mirror-target --description «Target dell'interfaccia di rete» -- eni-
eni-01f6f631e network-interface-id EXAMPLE
```

Output:

```

{
  "ClientToken": "5289a345-0358-4e62-93d5-47ef3061d65e",
  "TrafficMirrorTarget": {
    "Description": "Network interface target",
    "NetworkInterfaceId": "eni-01f6f631eEXAMPLE",
    "TrafficMirrorTargetId": "tmt-02dcdb2abEXAMPLE",
    "OwnerId": "111122223333",
    "Type": "network-interface",
    "Tags": []
  }
}

```

[Per ulteriori informazioni, consulta Creare un Traffic Mirror Target nella Traffic Mirroring Guide.AWS](#)

- Per API i dettagli, consulta [CreateTrafficMirrorTarget AWS CLI Command Reference](#).

## **create-transit-gateway-connect-peer**

Il seguente esempio di codice mostra come utilizzare `create-transit-gateway-connect-peer`.

AWS CLI

Per creare un peer Transit Gateway Connect

L'`create-transit-gateway-connect-peer` seguente crea un peer Connect.

```
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
```

Output:

```
{  
  "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"  
        }  
      ]  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [gli allegati Transit gateway Connect e i peer Transit Gateway Connect](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [CreateTransitGatewayConnectPeer](#) in AWS CLI Command Reference.

## create-transit-gateway-connect

Il seguente esempio di codice mostra come utilizzare `create-transit-gateway-connect`.

### AWS CLI

Per creare un gateway di transito (allegato Connect)

L'esempio seguente crea un allegato Connect, con il protocollo «gre», per l'allegato specificato.

```
aws ec2 create-transit-gateway-connect \
  --transport-transit-gateway-attachment-id tgw-attach-0a89069f57EXAMPLE \
  --options "Protocol=gre"
```

Output:

```
{
  "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"
    }
  }
}
```

Per ulteriori informazioni, consulta [gli allegati Transit gateway Connect e i peer Transit Gateway Connect](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [CreateTransitGatewayConnect](#) in AWS CLI Command Reference.

## create-transit-gateway-multicast-domain

Il seguente esempio di codice mostra come utilizzare `create-transit-gateway-multicast-domain`.

## AWS CLI

### Esempio 1: per creare un dominio IGMP multicast

L'`create-transit-gateway-multicast-domain` seguente crea un dominio multicast per il gateway di transito specificato. Con le sorgenti statiche disattivate, qualsiasi istanza nelle sottoreti associate al dominio multicast può inviare traffico multicast. Se almeno un membro utilizza il IGMP protocollo, è necessario abilitare il supporto. IGMPv2

```
aws ec2 create-transit-gateway-multicast-domain \  
  --transit-gateway-id tgw-0bf0bffefaEXAMPLE \  
  --options StaticSourcesSupport=disable,Igmpv2Support=enable
```

Output:

```
{  
  "TransitGatewayMulticastDomain": {  
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c9e29e2a7EXAMPLE",  
    "TransitGatewayId": "tgw-0bf0bffefaEXAMPLE",  
    "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-  
west-2:123456789012:transit-gateway-multicast-domain/tgw-mcast-  
domain-0c9e29e2a7EXAMPLE",  
    "OwnerId": "123456789012",  
    "Options": {  
      "Igmpv2Support": "enable",  
      "StaticSourcesSupport": "disable",  
      "AutoAcceptSharedAssociations": "disable"  
    },  
    "State": "pending",  
    "CreationTime": "2021-09-29T22:17:13.000Z"  
  }  
}
```

### Esempio 2: per creare un dominio multicast statico

L'`create-transit-gateway-multicast-domain` seguente crea un dominio multicast per il gateway di transito specificato. Con le sorgenti statiche abilitate, è necessario aggiungere fonti staticamente.

```
aws ec2 create-transit-gateway-multicast-domain \  
  --transit-gateway-id tgw-0bf0bffefaEXAMPLE \  
  --options StaticSourcesSupport=enable
```

```
--options StaticSourcesSupport=enable,Igmpv2Support=disable
```

Output:

```
{
  "TransitGatewayMulticastDomain": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-000fb24d04EXAMPLE",
    "TransitGatewayId": "tgw-0bf0bffe04EXAMPLE",
    "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-
west-2:123456789012:transit-gateway-multicast-domain/tgw-mcast-
domain-000fb24d04EXAMPLE",
    "OwnerId": "123456789012",
    "Options": {
      "Igmpv2Support": "disable",
      "StaticSourcesSupport": "enable",
      "AutoAcceptSharedAssociations": "disable"
    },
    "State": "pending",
    "CreationTime": "2021-09-29T22:20:19.000Z"
  }
}
```

Per ulteriori informazioni, consulta la sezione [Gestione dei domini multicast](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [CreateTransitGatewayMulticastDomain](#) in AWS CLI Command Reference.

## **create-transit-gateway-peering-attachment**

Il seguente esempio di codice mostra come utilizzare `create-transit-gateway-peering-attachment`.

AWS CLI

Per creare un allegato di peering del gateway di transito

L'esempio seguente crea una richiesta di peering attachment tra i due gateway di transito specificati.

```
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
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Transit Gateway Peering Attachments](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [CreateTransitGatewayPeeringAttachment](#) in AWS CLI Command Reference.

## create-transit-gateway-policy-table

Il seguente esempio di codice mostra come utilizzare `create-transit-gateway-policy-table`.

AWS CLI

Per creare una tabella di policy sui gateway di transito

L'esempio seguente crea una tabella di policy del gateway di transito per il gateway di transito specificato.

```
aws ec2 create-transit-gateway-policy-table \  

```

```
--transit-gateway-id tgw-067f8505c18f0bd6e
```

Output:

```
{
  "TransitGatewayPolicyTable": {
    "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
    "TransitGatewayId": "tgw-067f8505c18f0bd6e",
    "State": "pending",
    "CreationTime": "2023-11-28T16:36:43+00:00"
  }
}
```

Per ulteriori informazioni, consulta le [tabelle delle policy del Transit Gateway](#) nella Transit Gateway User Guide.

- Per API i dettagli, vedere [CreateTransitGatewayPolicyTable](#) in AWS CLI Command Reference.

## create-transit-gateway-prefix-list-reference

Il seguente esempio di codice mostra come utilizzare `create-transit-gateway-prefix-list-reference`.

### AWS CLI

Per creare un riferimento a un elenco di prefissi

L'esempio seguente crea un riferimento all'elenco di prefissi specificato nella tabella delle rotte del gateway di transito specificata.

```
aws ec2 create-transit-gateway-prefix-list-reference \
  --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
  --prefix-list-id pl-1111112222222333 \
  --transit-gateway-attachment-id tgw-attach-aaaaaabbbbb11111
```

Output:

```
{
  "TransitGatewayPrefixListReference": {
    "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
```



```

    "PrefixListId": "pl-11111122222222333",
    "PrefixListOwnerId": "123456789012",
    "State": "pending",
    "Blackhole": false,
    "TransitGatewayAttachment": {
      "TransitGatewayAttachmentId": "tgw-attach-aaaaaabbbbbb11111",
      "ResourceType": "vpc",
      "ResourceId": "vpc-112233445566aabbcc"
    }
  }
}

```

Per ulteriori informazioni, vedete i [riferimenti all'elenco dei prefissi](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [CreateTransitGatewayPrefixListReference](#) in AWS CLI Command Reference.

## create-transit-gateway-route-table

Il seguente esempio di codice mostra come utilizzare `create-transit-gateway-route-table`.

### AWS CLI

Per creare una tabella delle rotte del Transit Gateway

L'`create-transit-gateway-route-table` esempio seguente crea una tabella di routing per il gateway di transito specificato.

```

aws ec2 create-transit-gateway-route-table \
  --transit-gateway-id tgw-0262a0e521EXAMPLE

```

Output:

```

{
  "TransitGatewayRouteTable": {
    "TransitGatewayRouteTableId": "tgw-rtb-0960981be7EXAMPLE",
    "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
    "State": "pending",
    "DefaultAssociationRouteTable": false,
    "DefaultPropagationRouteTable": false,
    "CreationTime": "2019-07-10T19:01:46.000Z"
  }
}

```

```
}
```

Per ulteriori informazioni, consulta [Create a transit gateway table](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [CreateTransitGatewayRouteTable](#) in AWS CLI Command Reference.

## create-transit-gateway-route

Il seguente esempio di codice mostra come utilizzare `create-transit-gateway-route`.

### AWS CLI

Per creare un percorso di transito

L'esempio seguente crea una rotta, con la destinazione specificata, per la tabella delle rotte specificata.

```
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
```

Output:

```
{
  "Route": {
    "DestinationCidrBlock": "10.0.2.0/24",
    "TransitGatewayAttachments": [
      {
        "ResourceId": "vpc-0065acced4EXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
        "ResourceType": "vpc"
      }
    ],
    "Type": "static",
    "State": "active"
  }
}
```

Per ulteriori informazioni, consulta le [tabelle di routing dei gateway Transit](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [CreateTransitGatewayRoute](#) in AWS CLI Command Reference.

## create-transit-gateway-vpc-attachment

Il seguente esempio di codice mostra come utilizzare `create-transit-gateway-vpc-attachment`.

### AWS CLI

Esempio 1: associare un gateway di transito a VPC

L'esempio seguente crea un gateway di transito allegato al valore specificato VPC.

```
aws ec2 create-transit-gateway-vpc-attachment \
  --transit-gateway-id tgw-0262a0e521EXAMPLE \
  --vpc-id vpc-07e8ffd50f49335df \
  --subnet-id subnet-0752213d59EXAMPLE
```

Output:

```
{
  "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"
    }
  }
}
```

Per ulteriori informazioni, vedete [Creare un gateway di transito allegato a un VPC](#) nella Transit Gateways Guide.

Esempio 2: associare un gateway di transito a più sottoreti in un VPC

L'create-transit-gateway-vpc-attachmentesempio seguente crea un gateway di transito collegato alle sottoreti VPC e e alle sottoreti specificate.

```
aws ec2 create-transit-gateway-vpc-attachment \  
  --transit-gateway-id tgw-02f776b1a7EXAMPLE \  
  --vpc-id vpc-3EXAMPLE \  
  --subnet-ids "subnet-dEXAMPLE" "subnet-6EXAMPLE"
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, vedete [Creare un gateway di transito allegato a un VPC](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [CreateTransitGatewayVpcAttachment](#) in AWS CLI Command Reference.

## create-transit-gateway

Il seguente esempio di codice mostra come utilizzare create-transit-gateway.

### AWS CLI

Per creare un gateway di transito

L'`create-transit-gateway` seguente crea un gateway di transito.

```
aws ec2 create-transit-gateway \
  --description MyTGW \
  --
options AmazonSideAsn=64516,AutoAcceptSharedAttachments=enable,DefaultRouteTableAssociation=
```

Output:

```
{
  "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"
    }
  }
}
```

Per ulteriori informazioni, consulta [Creare un gateway di transito](#) nella Guida ai gateway di transito.

- Per API i dettagli, vedere [CreateTransitGateway](#) in AWS CLI Command Reference.

## **create-verified-access-endpoint**

Il seguente esempio di codice mostra come utilizzare `create-verified-access-endpoint`.

AWS CLI

Per creare un endpoint con accesso verificato

L'create-verified-access-endpointesempio seguente crea un endpoint Verified Access per il gruppo Verified Access specificato. L'interfaccia di rete e il gruppo di sicurezza specificati devono appartenere allo stesso VPC

```
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}]
```

Output:

```
{
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta gli [endpoint di accesso verificato](#) nella Guida per l'utente di AWS Verified Access.

- Per API i dettagli, consulta [CreateVerifiedAccessEndpoint AWS CLI Command Reference](#).

## create-verified-access-group

Il seguente esempio di codice mostra come utilizzare `create-verified-access-group`.

### AWS CLI

Per creare un gruppo con accesso verificato

L'`create-verified-access-group` seguente crea un gruppo di accesso verificato per l'istanza di accesso verificato specificata.

```

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}]

```

Output:

```

{
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta i [gruppi di accesso verificato](#) nella Guida per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [CreateVerifiedAccessGroup AWS CLI Command Reference](#).

## create-verified-access-instance

Il seguente esempio di codice mostra come utilizzare `create-verified-access-instance`.

### AWS CLI

Per creare un'istanza di accesso verificato

L'`create-verified-access-instance` esempio seguente crea un'istanza di accesso verificato con un tag Name.

```

aws ec2 create-verified-access-instance \
  --tag-specifications ResourceType=verified-access-
instance, Tags=[{Key=Name, Value=my-va-instance}]

```

Output:

```

{
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta le [istanze di accesso verificato](#) nella Guida per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [CreateVerifiedAccessInstance AWS CLI Command Reference](#).

## create-verified-access-trust-provider

Il seguente esempio di codice mostra come utilizzare `create-verified-access-trust-provider`.

### AWS CLI

Per creare un provider fiduciario di accesso verificato

L'esempio seguente configura un provider fiduciario di accesso verificato utilizzando AWS Identity Center.

```

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}]

```

Output:

```

{
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Trust providers for Verified Access](#) nella AWS Verified Access User Guide.

- Per API i dettagli, consulta [CreateVerifiedAccessTrustProvider AWS CLI Command Reference](#).

## create-volume

Il seguente esempio di codice mostra come utilizzare `create-volume`.

### AWS CLI

Per creare un volume General Purpose SSD (gp2) vuoto

L'esempio seguente crea un volume General Purpose SSD (gp2) da 80 GiB nella zona di disponibilità specificata. Si noti che la regione corrente deve essere `us-east-1`, oppure è possibile aggiungere il `--region` parametro per specificare la regione per il comando.

```

aws ec2 create-volume \
  --volume-type gp2 \
  --size 80 \
  --availability-zone us-east-1a

```

Output:

```

{
  "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
  }

```

Se non si specifica un tipo di volume, il tipo di volume predefinito è gp2.

```

aws ec2 create-volume \
  --size 80 \
  --availability-zone us-east-1a

```

Esempio 2: creare un volume Provisioned IOPS SSD (io1) da un'istantanea

L'create-volume esempio seguente crea un volume Provisioned IOPS SSD (io1) con 1000 provisioning IOPS nella zona di disponibilità specificata utilizzando l'istantanea specificata.

```

aws ec2 create-volume \
  --volume-type io1 \
  --iops 1000 \
  --snapshot-id snap-066877671789bd71b \
  --availability-zone us-east-1a

```

Output:

```

{
  "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
}

```

Esempio 3: creare un volume crittografato

L'create-volume esempio seguente crea un volume crittografato utilizzando l'impostazione predefinita CMK per la EBS crittografia. Se la crittografia per impostazione predefinita è disabilitata, è necessario specificare il `--encrypted` parametro come segue.

```
aws ec2 create-volume \  
  --size 80 \  
  --encrypted \  
  --availability-zone us-east-1a
```

Output:

```
{  
  "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  
}
```

Se la crittografia è abilitata per impostazione predefinita, il comando di esempio seguente crea un volume crittografato, anche senza il `--encrypted` parametro.

```
aws ec2 create-volume \  
  --size 80 \  
  --availability-zone us-east-1a
```

Se si utilizza il `--kms-key-id` parametro per specificare un servizio gestito dal cliente CMK, è necessario specificare il `--encrypted` parametro anche se la crittografia è abilitata per impostazione predefinita.

```
aws ec2 create-volume \  
  --volume-type gp2 \  
  --size 80 \  
  --encrypted \  
  --kms-key-id 0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE \  
  --availability-zone us-east-1a
```

```
--availability-zone us-east-1a
```

Esempio 4: creare un volume con tag

L'`create-volume` seguente crea un volume e aggiunge due tag.

```
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}]'
```

- Per API i dettagli, vedere [CreateVolume](#) in AWS CLI Command Reference.

## `create-vpc-endpoint-connection-notification`

Il seguente esempio di codice mostra come utilizzare `create-vpc-endpoint-connection-notification`.

### AWS CLI

Per creare una notifica di connessione all'endpoint

Questo esempio crea una notifica per un servizio endpoint specifico che avvisa l'utente quando gli endpoint dell'interfaccia si sono connessi al servizio e quando gli endpoint sono stati accettati per il servizio.

Comando:

```
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
```

Output:

```
{  
  "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"
    }
}

```

- Per i API dettagli, vedere [CreateVpcEndpointConnectionNotification](#) in Command Reference.AWS CLI

## create-vpc-endpoint-service-configuration

Il seguente esempio di codice mostra come utilizzare `create-vpc-endpoint-service-configuration`.

### AWS CLI

Esempio 1: creare una configurazione del servizio endpoint per un endpoint di interfaccia

L'`create-vpc-endpoint-service-configuration`esempio seguente crea una configurazione del servizio VPC endpoint utilizzando Network Load Balancer `nlb-vpce`. Questo esempio specifica inoltre che le richieste di connessione al servizio tramite un endpoint di interfaccia devono essere accettate.

```

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

```

Output:

```

{
  "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"
  ]
}
}

```

Esempio 2: creare una configurazione del servizio endpoint per un endpoint Gateway Load Balancer

L'create-vpc-endpoint-service-configurationesempio seguente crea una configurazione del servizio VPC endpoint utilizzando il Gateway Load BalancerGWLBService. Le richieste di connessione al servizio tramite un endpoint Gateway Load Balancer vengono accettate automaticamente.

```

aws ec2 create-vpc-endpoint-service-configuration \
  --gateway-load-balancer-arns arn:aws:elasticloadbalancing:us-
east-1:123456789012:loadbalancer/gwy/GWLBService/123123123123abcc \
  --no-acceptance-required

```

Output:

```

{
  "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/
        GWLBService/123123123123abcc"
    ]
}
}

```

Per ulteriori informazioni, consulta [i servizi VPC endpoint](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [CreateVpcEndpointServiceConfiguration AWS CLI Command Reference](#).

## create-vpc-endpoint

Il seguente esempio di codice mostra come utilizzare `create-vpc-endpoint`.

### AWS CLI

Esempio 1: per creare un endpoint gateway

L'`create-vpc-endpoint` seguente crea un VPC endpoint gateway tra Amazon S3 VPC `vpc-1a2b3c4d` e Amazon S3 nella regione e associa `us-east-1` la `rtb-11aa22bb` tabella di routing all'endpoint.

```

aws ec2 create-vpc-endpoint \
  --vpc-id vpc-1a2b3c4d \
  --service-name com.amazonaws.us-east-1.s3 \
  --route-table-ids rtb-11aa22bb

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Creazione di un endpoint gateway](#) nella Guida.AWS PrivateLink

### Esempio 2: creare un endpoint di interfaccia

L'create-vpc-endpointesempio seguente crea un VPC endpoint di interfaccia tra Amazon S3 VPC vpc-1a2b3c4d e Amazon S3 nella us-east-1 regione. Il comando crea l'endpoint nella sottoretesubnet-1a2b3c4d, lo associa al gruppo sg-1a2b3c4d di sicurezza e aggiunge un tag con una chiave «Service» e un valore di «S3».

```

aws ec2 create-vpc-endpoint \
  --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}]

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, vedere [Creazione di un endpoint di interfaccia](#) nella Guida per l'utente di AWS PrivateLink

### Esempio 3: creare un endpoint Gateway Load Balancer

L'create-vpc-endpointesempio seguente crea un endpoint Gateway Load Balancer tra VPC vpc-111122223333aabbcc e e un servizio configurato utilizzando un Gateway Load Balancer.

```

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
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta gli [endpoint Gateway Load Balancer nella Guida](#) per l'utente per AWS PrivateLink

- Per API i dettagli, vedere [CreateVpcEndpoint](#) in AWS CLI Command Reference.

## create-vpc-peering-connection

Il seguente esempio di codice mostra come utilizzare `create-vpc-peering-connection`.

### AWS CLI

Per creare una connessione VPC peering tra i tuoi VPCs

Questo esempio richiede una connessione peering tra `vpc-1a2b3c4d` e VPCs `vpc-11122233`.

Comando:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-  
id vpc-11122233
```

## Output:

```
{
  "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",
    "AcceptorVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-11122233"
    }
  }
}
```

Per creare VPC una VPC connessione peering con un altro account

Questo esempio richiede una connessione peering tra il tuo VPC (vpc-1a2b3c4d) e un (vpc-11122233) che appartiene all'account 123456789012. VPC AWS

Comando:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-id vpc-11122233 --peer-owner-id 123456789012
```

Per creare VPC una connessione peering con un in un'altra VPC regione

Questo esempio richiede una connessione peering tra l'utente VPC nella regione corrente (vpc-1a2b3c4d) e un (vpc-11122233) nel proprio account nella regione. VPC us-west-2

Comando:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-id vpc-11122233 --peer-region us-west-2
```

Questo esempio richiede una connessione peering tra il tuo account VPC nella regione corrente (vpc-1a2b3c4d) e un account (vpc-11122233) che appartiene all'account 123456789012 che si trova nella regione. VPC AWS us-west-2

Comando:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-id vpc-11122233 --peer-owner-id 123456789012 --peer-region us-west-2
```

- API Per [CreateVpcPeeringConnection AWS CLI](#) dettagli, vedere in Command Reference.

## create-vpc

Il seguente esempio di codice mostra come utilizzare `create-vpc`.

### AWS CLI

Esempio 1: Per creare un VPC

L'esempio seguente crea un tag VPC con il IPv4 CIDR blocco specificato e un tag Name.

```
aws ec2 create-vpc \
  --cidr-block 10.0.0.0/16 \
  --tag-specifications ResourceType=vpc, Tags=[{Key=Name, Value=MyVpc}]
```

Output:

```
{
  "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"
        }
    ]
}

```

## Esempio 2: creare una locazione VPC con locazione dedicata

L'create-vpcesempio seguente crea un VPC con il IPv4 CIDR blocco specificato e una locazione dedicata.

```

aws ec2 create-vpc \
  --cidr-block 10.0.0.0/16 \
  --instance-tenancy dedicated

```

Output:

```

{
  "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
  }
}

```

### Esempio 3: creare un blocco VPC con un IPv6 CIDR

L'create-vpcesempio seguente crea un VPC con un blocco fornito da Amazon IPv6CIDR.

```

aws ec2 create-vpc \
  --cidr-block 10.0.0.0/16 \
  --amazon-provided-ipv6-cidr-block

```

### Output:

```

{
  "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"
        }
      },
      {
        "AssociationId": "vpc-cidr-assoc-0669f8f9f5EXAMPLE",
        "CidrBlock": "10.0.0.0/16",
        "CidrBlockState": {
          "State": "associated"
        }
      }
    ]
  },
  "CidrBlockAssociationSet": [
    {
      "AssociationId": "vpc-cidr-assoc-0669f8f9f5EXAMPLE",
      "CidrBlock": "10.0.0.0/16",
      "CidrBlockState": {
        "State": "associated"
      }
    }
  ],
}

```

```

    "IsDefault": false
  }
}

```

#### Esempio 4: creare un oggetto VPC con un CIDR da un pool IPAM

L'create-vpcesempio seguente crea un VPC with a CIDR da un pool di Amazon VPC IP Address Manager (IPAM).

Linux e 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"}]
```

Output:

```

{
  "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"
      }
    ]
  }
}
```

Per ulteriori informazioni, consulta [Create a VPC that uses an IPAM pool CIDR](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [CreateVpc AWS CLI](#) Command Reference.

## **create-vpn-connection-route**

Il seguente esempio di codice mostra come utilizzare `create-vpn-connection-route`.

### AWS CLI

Per creare una route statica per una VPN connessione

Questo esempio crea una route statica per la VPN connessione specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 create-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-cidr-block 11.12.0.0/16
```

- Per API i dettagli, vedere [CreateVpnConnectionRoute](#) in AWS CLI Command Reference.

## **create-vpn-connection**

Il seguente esempio di codice mostra come utilizzare `create-vpn-connection`.

## AWS CLI

### Esempio 1: creare una VPN connessione con routing dinamico

L'create-vpn-connectionesempio seguente crea una VPN connessione tra il gateway privato virtuale specificato e il gateway del cliente specificato e applica i tag alla VPN connessione.

L'output include le informazioni di configurazione per il dispositivo gateway del cliente, in XML formato.

```
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}]'
```

Output:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

```

    ]
  }
}

```

Per ulteriori informazioni, consulta [How AWS Site-to-Site VPN works](#) nella Guida AWS Site-to-Site VPN per l'utente.

### Esempio 2: creare una VPN connessione con routing statico

L'`create-vpn-connection` seguente crea una VPN connessione tra il gateway privato virtuale specificato e il gateway del cliente specificato. Le opzioni specificano il routing statico. L'output include le informazioni di configurazione per il dispositivo gateway del cliente, in XML formato.

```

aws ec2 create-vpn-connection \
  --type ipsec.1 \
  --customer-gateway-id cgw-001122334455aabbc \
  --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
  --options "{\"StaticRoutesOnly\":true}"

```

### Output:

```

{
  "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": []
  }
}

```

Per ulteriori informazioni, consulta [How AWS Site-to-Site VPN works](#) nella Guida AWS Site-to-Site VPN per l'utente.

Esempio 3: creare una VPN connessione e specificare la propria chiave interna CIDR e già condivisa

L'create-vpn-connectionesempio seguente crea una VPN connessione e specifica il CIDR blocco di indirizzi IP interno e una chiave precondivisa personalizzata per ogni tunnel. I valori specificati vengono restituiti nelle informazioni. CustomerGatewayConfiguration

```

aws ec2 create-vpn-connection \
  --type ipsec.1 \
  --customer-gateway-id cgw-001122334455aabbc \
  --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
  --options
  TunnelOptions='[{TunnelInsideCidr=169.254.12.0/30,PreSharedKey=ExamplePreSharedKey1},
{TunnelInsideCidr=169.254.13.0/30,PreSharedKey=ExamplePreSharedKey2}]'

```

Output:

```

{
  "VpnConnection": {
    "CustomerGatewayConfiguration": "..configuration information...",
    "CustomerGatewayId": "cgw-001122334455aabbc",
    "Category": "VPN",
    "State": "pending",
    "VpnConnectionId": "vpn-123123123123abcb",
    "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",
    "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": []
}
}

```

Per ulteriori informazioni, consulta [How AWS Site-to-Site VPN works](#) in the AWS Site-to-Site VPN User Guide.

Esempio 4: creare una VPN connessione che supporti IPv6 il traffico

L'create-vpn-connectionesempio seguente crea una VPN connessione che supporta il IPv6 traffico tra il gateway di transito specificato e il gateway del cliente specificato. Le opzioni del tunnel per entrambi i tunnel specificano chi AWS deve avviare la IKE negoziazione.

```

aws ec2 create-vpn-connection \
  --type ipsec.1 \
  --transit-gateway-id tgw-12312312312312312 \
  --customer-gateway-id cgw-001122334455aabbcc \
  --options TunnelInsideIpVersion=ipv6,TunnelOptions=[{StartupAction=start},
{StartupAction=start}]

```

Output:

```

{
  "VpnConnection": {
    "CustomerGatewayConfiguration": "..configuration information...",
    "CustomerGatewayId": "cgw-001122334455aabbcc",
    "Category": "VPN",
    "State": "pending",
    "VpnConnectionId": "vpn-111111111122222222",
    "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": []
  }
}

```

Per ulteriori informazioni, consulta [How AWS Site-to-Site VPN works](#) nella User Guide.AWS Site-to-Site VPN

- Per API i dettagli, vedere [CreateVpnConnection](#) in AWS CLI Command Reference.

## create-vpn-gateway

Il seguente esempio di codice mostra come utilizzare `create-vpn-gateway`.

### AWS CLI

Per creare un gateway privato virtuale

Questo esempio crea un gateway privato virtuale.

Comando:

```
aws ec2 create-vpn-gateway --type ipsec.1
```

Output:

```

{
  "VpnGateway": {
    "AmazonSideAsn": 64512,

```

```
    "State": "available",
    "Type": "ipsec.1",
    "VpnGatewayId": "vgw-9a4cacf3",
    "VpcAttachments": []
  }
}
```

Per creare un gateway privato virtuale con un lato Amazon specifico ASN

Questo esempio crea un gateway privato virtuale e specifica l'Autonomous System Number (ASN) per il lato Amazon della BGP sessione.

Comando:

```
aws ec2 create-vpn-gateway --type ipsec.1 --amazon-side-asn 65001
```

Output:

```
{
  "VpnGateway": {
    "AmazonSideAsn": 65001,
    "State": "available",
    "Type": "ipsec.1",
    "VpnGatewayId": "vgw-9a4cacf3",
    "VpcAttachments": []
  }
}
```

- Per API i dettagli, consulta [CreateVpnGateway AWS CLI Command Reference](#).

## **delete-carrier-gateway**

Il seguente esempio di codice mostra come utilizzare `delete-carrier-gateway`.

### AWS CLI

Per eliminare il gateway del tuo operatore

L'esempio seguente elimina il gateway dell'operatore telefonico specificato.

```
aws ec2 delete-carrier-gateway \
```

```
--carrier-gateway-id cagw-0465cdEXAMPLE1111
```

Output:

```
{
  "CarrierGateway": {
    "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
    "VpcId": "vpc-0c529aEXAMPLE1111",
    "State": "deleting",
    "OwnerId": "123456789012"
  }
}
```

Per ulteriori informazioni, consulta [Carrier Gateways](#) nella Amazon Virtual Private Cloud User Guide.

- Per API i dettagli, consulta [DeleteCarrierGateway AWS CLI Command Reference](#).

## delete-client-vpn-endpoint

Il seguente esempio di codice mostra come utilizzare `delete-client-vpn-endpoint`.

AWS CLI

Per eliminare un VPN endpoint Client

L'`delete-client-vpn-endpoint` seguente elimina l'endpoint Client VPN specificato.

```
aws ec2 delete-client-vpn-endpoint \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Output:

```
{
  "Status": {
    "Code": "deleting"
  }
}
```

Per ulteriori informazioni, consulta [Client VPN Endpoints nella Client Administrator AWS Guide](#).

VPN



- Per API i dettagli, vedere [DeleteClientVpnEndpoint](#) in AWS CLI Command Reference.

## delete-client-vpn-route

Il seguente esempio di codice mostra come utilizzare `delete-client-vpn-route`.

### AWS CLI

Per eliminare una route per un VPN endpoint Client

L'`delete-client-vpn-route` esempio seguente elimina la `0.0.0.0/0` route per la sottorete specificata di un endpoint Client. VPN

```
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
```

Output:

```
{
  "Status": {
    "Code": "deleting"
  }
}
```

Per ulteriori informazioni, vedere [Routes](#) nella AWS Client VPN Administrator Guide.

- Per API i dettagli, vedere [DeleteClientVpnRoute](#) in AWS CLI Command Reference.

## delete-coip-cidr

Il seguente esempio di codice mostra come utilizzare `delete-coip-cidr`.

### AWS CLI

Per eliminare un intervallo di indirizzi IP (CoIP) di proprietà del cliente

L'`delete-coip-cidr` esempio seguente elimina l'intervallo specificato di indirizzi CoIP nel pool CoIP specificato.

```
aws ec2 delete-coip-cidr \
```

```
--cidr 14.0.0.0/24 \  
--coip-pool-id ipv4pool-coip-1234567890abcdefg
```

Output:

```
{  
  "CoipCidr": {  
    "Cidr": "14.0.0.0/24",  
    "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",  
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"  
  }  
}
```

Per ulteriori informazioni, consulta [Indirizzi IP di proprietà del cliente](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, vedere [DeleteCoipCidr](#) in AWS CLI Command Reference.

## delete-coip-pool

Il seguente esempio di codice mostra come utilizzare `delete-coip-pool`.

AWS CLI

Per eliminare un pool di indirizzi IP (CoIP) di proprietà del cliente

L'esempio seguente elimina un pool CoIP di indirizzi CoIP.

```
aws ec2 delete-coip-pool \  
--coip-pool-id ipv4pool-coip-1234567890abcdefg
```

Output:

```
{  
  "CoipPool": {  
    "PoolId": "ipv4pool-coip-1234567890abcdefg",  
    "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",  
    "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-coip-1234567890abcdefg"  
  }  
}
```

Per ulteriori informazioni, consulta [Indirizzi IP di proprietà del cliente](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, vedere [DeleteCoipPool](#) in AWS CLI Command Reference.

## **delete-customer-gateway**

Il seguente esempio di codice mostra come utilizzare `delete-customer-gateway`.

### AWS CLI

Per eliminare un gateway per i clienti

Questo esempio elimina il Customer Gateway specificato. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-customer-gateway --customer-gateway-id cgw-0e11f167
```

- Per API i dettagli, vedere [DeleteCustomerGateway](#) in AWS CLI Command Reference.

## **delete-dhcp-options**

Il seguente esempio di codice mostra come utilizzare `delete-dhcp-options`.

### AWS CLI

Per eliminare un set DHCP di opzioni

Questo esempio elimina il set di DHCP opzioni specificato. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-dhcp-options --dhcp-options-id dopt-d9070ebb
```

- Per API i dettagli, vedere [DeleteDhcpOptions](#) in AWS CLI Command Reference.

## **delete-egress-only-internet-gateway**

Il seguente esempio di codice mostra come utilizzare `delete-egress-only-internet-gateway`.

## AWS CLI

Per eliminare un gateway Internet solo in uscita

Questo esempio elimina il gateway Internet di sola uscita specificato.

Comando:

```
aws ec2 delete-egress-only-internet-gateway --egress-only-internet-gateway-id eigw-01eadbd45ecd7943f
```

Output:

```
{
  "ReturnCode": true
}
```

- Per i API dettagli, vedere [DeleteEgressOnlyInternetGateway](#) in Command Reference.AWS CLI

## delete-fleets

Il seguente esempio di codice mostra come utilizzare `delete-fleets`.

### AWS CLI

Esempio 1: eliminare un EC2 parco istanze e terminare le istanze associate

L'`delete-fleets` esempio seguente elimina il EC2 parco istanze specificato e termina le istanze On-Demand e le istanze Spot associate.

```
aws ec2 delete-fleets \
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
  --terminate-instances
```

Output:

```
{
  "SuccessfulFleetDeletions": [
    {
      "CurrentFleetState": "deleted_terminating",
      "PreviousFleetState": "active",
      "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
    }
  ]
}
```

```
    }
  ],
  "UnsuccessfulFleetDeletions": []
}
```

Per ulteriori informazioni, consulta [Delete an EC2 Fleet](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

Esempio 2: eliminare una EC2 flotta senza terminare le istanze associate

L'example seguente elimina il parco istanze specificato senza terminare le istanze On-Demand e le istanze Spot associate.

```
aws ec2 delete-fleets \
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
  --no-terminate-instances
```

Output:

```
{
  "SuccessfulFleetDeletions": [
    {
      "CurrentFleetState": "deleted_running",
      "PreviousFleetState": "active",
      "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
    }
  ],
  "UnsuccessfulFleetDeletions": []
}
```

Per ulteriori informazioni, consulta [Delete an EC2 Fleet](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

- Per API i dettagli, consulta AWS CLI Command [DeleteFleets](#) Reference.

## delete-flow-logs

Il seguente esempio di codice mostra come utilizzare `delete-flow-logs`.

### AWS CLI

Per eliminare un log di flusso

L'`delete-flow-logs` seguente elimina il log di flusso specificato.

```
aws ec2 delete-flow-logs --flow-log-id fl-11223344556677889
```

Output:

```
{
  "Unsuccessful": []
}
```

- Per API i dettagli, vedere [DeleteFlowLogs](#) in AWS CLI Command Reference.

## **delete-fpga-image**

Il seguente esempio di codice mostra come utilizzare `delete-fpga-image`.

AWS CLI

Per eliminare un'FPGA immagine Amazon

Questo esempio elimina il file specificato AFI.

Comando:

```
aws ec2 delete-fpga-image --fpga-image-id afi-06b12350a123fbabc
```

Output:

```
{
  "Return": true
}
```

- Per API i dettagli, vedere [DeleteFpgaImage](#) in AWS CLI Command Reference.

## **delete-instance-connect-endpoint**

Il seguente esempio di codice mostra come utilizzare `delete-instance-connect-endpoint`.

AWS CLI

Per eliminare un endpoint EC2 Instance Connect

L'`delete-instance-connect-endpoint` seguente elimina l'endpoint EC2 Instance Connect specificato.

```
aws ec2 delete-instance-connect-endpoint \  
  --instance-connect-endpoint-id eice-03f5e49b83924bbc7
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Remove EC2 Instance Connect Endpoint](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DeleteInstanceConnectEndpoint AWS CLI Command Reference](#).

## **delete-instance-event-window**

Il seguente esempio di codice mostra come utilizzare `delete-instance-event-window`.

AWS CLI

Esempio 1: Per eliminare una finestra di eventi

L'`delete-instance-event-window` seguente elimina una finestra di eventi.

```
aws ec2 delete-instance-event-window \  
  --region us-east-1 \  
  --instance-id i-0123456789
```

```
--instance-event-window-id iew-0abcdef1234567890
```

Output:

```
{
  "InstanceEventWindowState": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "State": "deleting"
  }
}
```

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella sezione Eventi pianificati della Amazon EC2 User Guide.

Esempio 2: Per forzare l'eliminazione di una finestra di evento

L'`delete-instance-event-window` seguente elimina forzatamente una finestra di eventi se la finestra degli eventi è attualmente associata a destinazioni.

```
aws ec2 delete-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --force-delete
```

Output:

```
{
  "InstanceEventWindowState": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "State": "deleting"
  }
}
```

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella sezione Eventi pianificati della Amazon EC2 User Guide.

- Per API i dettagli, consulta Command [DeleteInstanceEventWindow](#) Reference AWS CLI .

## **delete-internet-gateway**

Il seguente esempio di codice mostra come utilizzare `delete-internet-gateway`.



## AWS CLI

Per eliminare un gateway Internet

L'`delete-internet-gateway` esempio seguente elimina il gateway Internet specificato.

```
aws ec2 delete-internet-gateway \  
  --internet-gateway-id igw-0d0fb496b3EXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Internet gateway](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [DeleteInternetGateway AWS CLI Command Reference](#).

## `delete-ipam-pool`

Il seguente esempio di codice mostra come utilizzare `delete-ipam-pool`.

## AWS CLI

Per eliminare un IPAM pool

In questo esempio, sei un amministratore IPAM delegato che desidera eliminare un IPAM pool che non ti serve più, ma al pool è stato assegnato un CIDR provisioning. Non puoi eliminare un pool a cui è stato assegnato il CIDRs provisioning a meno che tu non utilizzi l'`--cascade` opzione, quindi userai `--cascade`

Per completare questa richiesta:

Avrai bisogno dell'ID del IPAM pool con cui puoi ottenere [describe-ipam-pools](#). `--region` Deve essere la regione di IPAM origine.

L'`delete-ipam-pool` esempio seguente elimina un IPAM pool nel tuo AWS account.

```
aws ec2 delete-ipam-pool \  
  --ipam-pool-id ipam-pool-050c886a3ca41cd5b \  
  --cascade \  
  --region us-east-1
```

Output:

```
{
```

```
"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
}
```

Per ulteriori informazioni, consulta [Eliminare un pool](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [DeleteIpamPool AWS CLI Command Reference](#).

## **delete-ipam-resource-discovery**

Il seguente esempio di codice mostra come utilizzare `delete-ipam-resource-discovery`.

### AWS CLI

Per eliminare una scoperta di risorse

In questo esempio, sei un amministratore IPAM delegato che desidera eliminare un'individuazione di risorse non predefinita che hai creato per condividerla con un altro IPAM amministratore durante il processo di integrazione IPAM con account esterni alla tua organizzazione.

Per completare questa richiesta:

`--region` Deve essere la regione in cui è stata creata l'individuazione delle risorse. Non è possibile eliminare un'individuazione di risorse predefinita se `"IsDefault": true`. Un'individuazione di risorse predefinita è quella che viene creata automaticamente nell'account

che crea un IPAM. Per eliminare un rilevamento di risorse predefinito, è necessario eliminare il IPAM.

L'esempio seguente elimina una scoperta di risorse.

```
aws ec2 delete-ipam-resource-discovery \
  --ipam-resource-discovery-id ipam-res-disco-0e39761475298ee0f \
  --region us-east-1
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni sulle scoperte di risorse, consulta [Work with resource discoveries](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteIpamResourceDiscovery](#) Reference.

## delete-ipam-scope

Il seguente esempio di codice mostra come utilizzare `delete-ipam-scope`.

AWS CLI

Per eliminare un IPAM ambito

L'esempio seguente elimina un IPAM.

```
aws ec2 delete-ipam-scope \  
  --ipam-scope-id ipam-scope-01c1ebab2b63bd7e4
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Eliminare un ambito](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [DeletelIpamScope AWS CLI Command Reference](#).

## delete-ipam

Il seguente esempio di codice mostra come utilizzare `delete-ipam`.

### AWS CLI

Per eliminare un IPAM

L'esempio seguente elimina un IPAM.

```
aws ec2 delete-ipam \  
  --ipam-id ipam-036486dfa6af58ee0
```

Output:

```
{  
  "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"
}
```

Per ulteriori informazioni, consulta [Delete an IPAM](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [DeleteIpam AWS CLI Command Reference](#).

## delete-key-pair

Il seguente esempio di codice mostra come utilizzare `delete-key-pair`.

### AWS CLI

Per eliminare una coppia di chiavi

L'esempio seguente elimina la coppia di chiavi specificata.

```
aws ec2 delete-key-pair \
  --key-name my-key-pair
```

Output:

```
{
  "Return": true,
```

```
"KeyPairId": "key-03c8d3aceb53b507"
}
```

Per ulteriori informazioni, vedete [Creare ed eliminare coppie di chiavi](#) nella Guida per l'utente dell'interfaccia a riga di AWS comando.

- Per API i dettagli, vedere [DeleteKeyPair](#) in AWS CLI Command Reference.

## delete-launch-template-versions

Il seguente esempio di codice mostra come utilizzare `delete-launch-template-versions`.

### AWS CLI

Per eliminare una versione del modello di avvio

Questo esempio elimina la versione del modello di lancio specificata.

Comando:

```
aws ec2 delete-launch-template-versions --launch-template-id lt-0abcd290751193123 --  
versions 1
```

Output:

```
{  
  "UnsuccessfullyDeletedLaunchTemplateVersions": [],  
  "SuccessfullyDeletedLaunchTemplateVersions": [  
    {  
      "LaunchTemplateName": "TestVersion",  
      "VersionNumber": 1,  
      "LaunchTemplateId": "lt-0abcd290751193123"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [DeleteLaunchTemplateVersions](#) in AWS CLI Command Reference.

## delete-launch-template

Il seguente esempio di codice mostra come utilizzare `delete-launch-template`.

## AWS CLI

Per eliminare un modello di avvio

In questo esempio viene eliminato il modello di avvio specificato.

Comando:

```
aws ec2 delete-launch-template --launch-template-id lt-0abcd290751193123
```

Output:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 2,
    "LaunchTemplateId": "lt-0abcd290751193123",
    "LaunchTemplateName": "TestTemplate",
    "DefaultVersionNumber": 2,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-11-23T16:46:25.000Z"
  }
}
```

- Per API i dettagli, vedere [DeleteLaunchTemplate](#) in AWS CLI Command Reference.

## **delete-local-gateway-route-table-virtual-interface-group-association**

Il seguente esempio di codice mostra come utilizzare `delete-local-gateway-route-table-virtual-interface-group-association`.

## AWS CLI

Per dissociare una tabella di routing del gateway locale da un gruppo di interfacce virtuali (VIFs)

L'`delete-local-gateway-route-table-virtual-interface-group-association` esempio seguente elimina l'associazione tra la tabella di routing del gateway locale e il gruppo specificati. VIF

```
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
```

**Output:**

```
{
  "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": []
  }
}
```

Per ulteriori informazioni, consulta le [associazioni di VIF gruppo](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, consulta [DeleteLocalGatewayRouteTableVirtualInterfaceGroupAssociation AWS CLI Command Reference](#).

**delete-local-gateway-route-table-vpc-association**

Il seguente esempio di codice mostra come utilizzare `delete-local-gateway-route-table-vpc-association`.

**AWS CLI**

Per dissociare una tabella di routing del gateway locale da una VPC

L'`delete-local-gateway-route-table-vpc-association` seguente elimina l'associazione tra la tabella di routing del gateway locale specificata e VPC

```
aws ec2 delete-local-gateway-route-table-vpc-association \
  --local-gateway-route-table-vpc-association-id vpc-example0123456789
```

**Output:**

```
{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [VPCle associazioni](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, consulta [DeleteLocalGatewayRouteTableVpcAssociation AWS CLI Command Reference](#).

## delete-local-gateway-route-table

Il seguente esempio di codice mostra come utilizzare `delete-local-gateway-route-table`.

### AWS CLI

Per eliminare una tabella di routing del gateway locale

L'esempio seguente crea una tabella di routing del gateway locale con la modalità di VPC routing diretto.

```

aws ec2 delete-local-gateway-route-table \
  --local-gateway-route-table-id lgw-rtb-abcdefg1234567890

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Local gateway route tables](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, vedere [DeleteLocalGatewayRouteTable](#) in AWS CLI Command Reference.

## delete-local-gateway-route

Il seguente esempio di codice mostra come utilizzare `delete-local-gateway-route`.

### AWS CLI

Per eliminare una route da una tabella di routing del gateway locale

L'esempio seguente elimina la route specificata dalla tabella di route del gateway locale specificata.

```

aws ec2 delete-local-gateway-route \
  --destination-cidr-block 0.0.0.0/0 \
  --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE

```

Output:

```

{
  "Route": {
    "DestinationCidrBlock": "0.0.0.0/0",
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
    "Type": "static",
    "State": "deleted",
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"
  }
}

```

- Per API i dettagli, vedere [DeleteLocalGatewayRoute](#) in AWS CLI Command Reference.

## delete-managed-prefix-list

Il seguente esempio di codice mostra come utilizzare `delete-managed-prefix-list`.

## AWS CLI

Per eliminare un elenco di prefissi

L'delete-managed-prefix-listesempio seguente elimina l'elenco di prefissi specificato.

```
aws ec2 delete-managed-prefix-list \  
  --prefix-list-id pl-0123456abcabc1
```

Output:

```
{  
  "PrefixList": {  
    "PrefixListId": "pl-0123456abcabc1",  
    "AddressFamily": "IPv4",  
    "State": "delete-in-progress",  
    "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/  
pl-0123456abcabc1",  
    "PrefixListName": "test",  
    "MaxEntries": 10,  
    "Version": 1,  
    "OwnerId": "123456789012"  
  }  
}
```

Per ulteriori informazioni, consulta [Managed Prefix Lists](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [DeleteManagedPrefixList AWS CLI](#) Command Reference.

## delete-nat-gateway

Il seguente esempio di codice mostra come utilizzare delete-nat-gateway.

### AWS CLI

Per eliminare un NAT gateway

Questo esempio elimina il NAT gateway `nat-04ae55e711cec5680`.

Comando:

```
aws ec2 delete-nat-gateway --nat-gateway-id nat-04ae55e711cec5680
```

## Output:

```
{
  "NatGatewayId": "nat-04ae55e711cec5680"
}
```

- Per API i dettagli, vedere [DeleteNatGateway](#) in AWS CLI Command Reference.

## delete-network-acl-entry

Il seguente esempio di codice mostra come utilizzare `delete-network-acl-entry`.

### AWS CLI

Per eliminare una ACL voce di rete

Questo esempio elimina la regola di ingresso numero 100 dalla rete specificata. ACL Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100
```

- Per API i dettagli, vedere [DeleteNetworkAclEntry](#) in AWS CLI Command Reference.

## delete-network-acl

Il seguente esempio di codice mostra come utilizzare `delete-network-acl`.

### AWS CLI

Per eliminare una rete ACL

Questo esempio elimina la rete ACL specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-network-acl --network-acl-id acl-5fb85d36
```

- Per API i dettagli, vedere [DeleteNetworkAcl](#) in AWS CLI Command Reference.

## delete-network-insights-access-scope-analysis

Il seguente esempio di codice mostra come utilizzare `delete-network-insights-access-scope-analysis`.

### AWS CLI

Per eliminare un'analisi dell'ambito di accesso alla rete

L'`delete-network-insights-access-scope-analysis` seguente elimina l'analisi dell'ambito di accesso alla rete specificata.

```
aws ec2 delete-network-insights-access-scope-analysis \
  --network-insights-access-scope-analysis-id nisa-01234567891abcdef
```

Output:

```
{
  "NetworkInsightsAccessScopeAnalysisId": "nisa-01234567891abcdef"
}
```

Per ulteriori informazioni, vedere [Guida introduttiva all'uso di Network Access Analyzer utilizzando la guida AWS CLI Network Access Analyzer](#).

- Per API i dettagli, vedere [DeleteNetworkInsightsAccessScopeAnalysis](#) in AWS CLI Command Reference.

## delete-network-insights-access-scope

Il seguente esempio di codice mostra come utilizzare `delete-network-insights-access-scope`.

### AWS CLI

Per eliminare un ambito di accesso alla rete

L'`delete-network-insights-access-scope` seguente elimina l'ambito di accesso alla rete specificato.

```
aws ec2 delete-network-insights-access-scope \  
--network-insights-access-scope-id nis-123456789abc01234
```

Output:

```
{  
  "NetworkInsightsAccessScopeId": "nis-123456789abc01234"  
}
```

Per ulteriori informazioni, vedere [Guida introduttiva all'uso di Network Access Analyzer utilizzando la guida AWS CLI Network Access Analyzer](#).

- Per API i dettagli, vedere [DeleteNetworkInsightsAccessScope](#) in AWS CLI Command Reference.

## delete-network-insights-analysis

Il seguente esempio di codice mostra come utilizzare `delete-network-insights-analysis`.

AWS CLI

Per eliminare un'analisi del percorso

L'`delete-network-insights-analysis` seguente elimina l'analisi specificata.

```
aws ec2 delete-network-insights-analysis \  
--network-insights-analysis-id nia-02207aa13eb480c7a
```

Output:

```
{  
  "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a"  
}
```

Per ulteriori informazioni, consulta la [Guida introduttiva al AWS CLI Reachability Analyzer Guide](#).

- Per API i dettagli, vedere [DeleteNetworkInsightsAnalysis](#) in AWS CLI Command Reference.

## delete-network-insights-path

Il seguente esempio di codice mostra come utilizzare `delete-network-insights-path`.

## AWS CLI

Per eliminare un percorso

L'`delete-network-insights-path` seguente elimina il percorso specificato. Prima di poter eliminare un percorso, è necessario eliminare tutte le relative analisi utilizzando il `delete-network-insights-analysis` comando.

```
aws ec2 delete-network-insights-path \  
  --network-insights-path-id nip-0b26f224f1d131fa8
```

Output:

```
{  
  "NetworkInsightsPathId": "nip-0b26f224f1d131fa8"  
}
```

Per ulteriori informazioni, consulta la [Guida introduttiva al AWS CLI Reachability Analyzer Guide](#).

- Per API i dettagli, vedere [DeleteNetworkInsightsPath](#) in AWS CLI Command Reference.

## `delete-network-interface-permission`

Il seguente esempio di codice mostra come utilizzare `delete-network-interface-permission`.

### AWS CLI

Per eliminare l'autorizzazione di un'interfaccia di rete

Questo esempio elimina l'autorizzazione dell'interfaccia di rete specificata.

Comando:

```
aws ec2 delete-network-interface-permission --network-interface-permission-id eni-  
perm-06fd19020ede149ea
```

Output:

```
{  
  "Return": true  
}
```

- Per API i dettagli, vedere [DeleteNetworkInterfacePermission](#) in AWS CLI Command Reference.

## **delete-network-interface**

Il seguente esempio di codice mostra come utilizzare `delete-network-interface`.

### AWS CLI

Per eliminare un'interfaccia di rete

Questo esempio elimina l'interfaccia di rete specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-network-interface --network-interface-id eni-e5aa89a3
```

- Per API i dettagli, vedere [DeleteNetworkInterface](#) in AWS CLI Command Reference.

## **delete-placement-group**

Il seguente esempio di codice mostra come utilizzare `delete-placement-group`.

### AWS CLI

Per eliminare un gruppo di posizionamenti

Questo comando di esempio elimina il gruppo di posizionamento specificato.

Comando:

```
aws ec2 delete-placement-group --group-name my-cluster
```

- Per API i dettagli, vedete [DeletePlacementGroup](#) in AWS CLI Command Reference.

## **delete-queued-reserved-instances**

Il seguente esempio di codice mostra come utilizzare `delete-queued-reserved-instances`.

### AWS CLI

Per eliminare un acquisto in coda



L'`delete-queued-reserved-instances` seguente elimina l'istanza riservata specificata, che era in coda per l'acquisto.

```
aws ec2 delete-queued-reserved-instances \
  --reserved-instances-ids af9f760e-6f91-4559-85f7-4980eexample
```

Output:

```
{
  "SuccessfulQueuedPurchaseDeletions": [
    {
      "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"
    }
  ],
  "FailedQueuedPurchaseDeletions": []
}
```

- Per API i dettagli, vedere [DeleteQueuedReservedInstances](#) in AWS CLI Command Reference.

## `delete-route-table`

Il seguente esempio di codice mostra come utilizzare `delete-route-table`.

AWS CLI

Per eliminare una tabella di percorsi

Questo esempio elimina la tabella di routing specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-route-table --route-table-id rtb-22574640
```

- Per API i dettagli, vedere [DeleteRouteTable](#) in AWS CLI Command Reference.

## `delete-route`

Il seguente esempio di codice mostra come utilizzare `delete-route`.

## AWS CLI

Per eliminare un percorso

Questo esempio elimina la rotta specificata dalla tabella delle rotte specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-route --route-table-id rtb-22574640 --destination-cidr-block 0.0.0.0/0
```

- Per API i dettagli, vedere [DeleteRoute](#) in AWS CLI Command Reference.

## delete-security-group

Il seguente esempio di codice mostra come utilizzare `delete-security-group`.

### AWS CLI

[EC2-Classical] Per eliminare un gruppo di sicurezza

In questo esempio viene eliminato il gruppo di sicurezza denominato `MySecurityGroup`. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-security-group --group-name MySecurityGroup
```

[EC2-VPC] Per eliminare un gruppo di sicurezza

In questo esempio viene eliminato il gruppo di sicurezza con ID `sg-903004f8`. Tieni presente che non puoi fare riferimento a un gruppo di sicurezza per EC2 - VPC per nome. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-security-group --group-id sg-903004f8
```

Per ulteriori informazioni, consulta [Utilizzo dei gruppi di sicurezza](#) nella Guida per l'utente dell'Interfaccia a riga di comando AWS .

- Per API i dettagli, vedere [DeleteSecurityGroup](#) in AWS CLI Command Reference.

## **delete-snapshot**

Il seguente esempio di codice mostra come utilizzare `delete-snapshot`.

### AWS CLI

Per eliminare uno snapshot

Questo comando di esempio elimina uno snapshot con ID `snap-1234567890abcdef0`. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-snapshot --snapshot-id snap-1234567890abcdef0
```

- Per API i dettagli, vedere [DeleteSnapshot](#) in AWS CLI Command Reference.

## **delete-spot-datafeed-subscription**

Il seguente esempio di codice mostra come utilizzare `delete-spot-datafeed-subscription`.

### AWS CLI

Per annullare un abbonamento al data feed di istanze Spot

Questo comando di esempio elimina un abbonamento al feed di dati Spot per l'account. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-spot-datafeed-subscription
```

- Per API i dettagli, consulta [DeleteSpotDatafeedSubscription AWS CLI](#) Command Reference.

## **delete-subnet-cidr-reservation**

Il seguente esempio di codice mostra come utilizzare `delete-subnet-cidr-reservation`.

## AWS CLI

Per eliminare una prenotazione di sottorete CIDR

L'example seguente elimina la prenotazione di CIDR sottorete specificata.

```
aws ec2 delete-subnet-cidr-reservation \  
  --subnet-cidr-reservation-id scr-044f977c4eEXAMPLE
```

Output:

```
{  
  "DeletedSubnetCidrReservation": {  
    "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",  
    "SubnetId": "subnet-03c51e2e6cEXAMPLE",  
    "Cidr": "10.1.0.16/28",  
    "ReservationType": "prefix",  
    "OwnerId": "123456789012"  
  }  
}
```

Per ulteriori informazioni, consulta [Subnet CIDR reservations](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [DeleteSubnetCidrReservation AWS CLI Command Reference](#).

## delete-subnet

Il seguente esempio di codice mostra come utilizzare `delete-subnet`.

### AWS CLI

Per eliminare una sottorete

Questo esempio elimina la sottorete specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-subnet --subnet-id subnet-9d4a7b6c
```

- Per API i dettagli, vedere [DeleteSubnet](#) in AWS CLI Command Reference.

## delete-tags

Il seguente esempio di codice mostra come utilizzare `delete-tags`.

### AWS CLI

Esempio 1: Per eliminare un tag da una risorsa

L'`delete-tags`esempio seguente elimina il tag `Stack=Test` dall'immagine specificata. Quando specificate sia un valore che un nome di chiave, il tag viene eliminato solo se il valore del tag corrisponde al valore specificato.

```
aws ec2 delete-tags \  
  --resources ami-1234567890abcdef0 \  
  --tags Key=Stack,Value=Test
```

È facoltativo specificare il valore per un tag. L'`delete-tags`esempio seguente elimina il tag con il nome della chiave `purpose` dall'istanza specificata, indipendentemente dal valore del tag.

```
aws ec2 delete-tags \  
  --resources i-1234567890abcdef0 \  
  --tags Key=purpose
```

Se specificate la stringa vuota come valore del tag, il tag viene eliminato solo se il valore del tag è la stringa vuota. L'`delete-tags`esempio seguente specifica la stringa vuota come valore del tag da eliminare.

```
aws ec2 delete-tags \  
  --resources i-1234567890abcdef0 \  
  --tags Key=Name,Value=
```

Esempio 2: Per eliminare un tag da più risorse

L'`delete-tags`esempio seguente elimina il tag `Purpose=test` sia da un'istanza che da un'AMI. Come mostrato nell'esempio precedente, è possibile omettere il valore del tag dal comando.

```
aws ec2 delete-tags \  
  --resources i-1234567890abcdef0 ami-1234567890abcdef0 \  
  --tags Key=Purpose
```

- Per API i dettagli, vedere [DeleteTags](#) in AWS CLI Command Reference.

## delete-traffic-mirror-filter-rule

Il seguente esempio di codice mostra come utilizzare `delete-traffic-mirror-filter-rule`.

### AWS CLI

Per eliminare una regola di filtro Traffic Mirror

L'`delete-traffic-mirror-filter-rule`esempio seguente elimina la regola del filtro Traffic Mirror specificata.

```
aws ec2 delete-traffic-mirror-filter-rule \  
  --traffic-mirror-filter-rule-id tmfr-081f71283bEXAMPLE
```

Output:

```
{  
  "TrafficMirrorFilterRuleId": "tmfr-081f71283bEXAMPLE"  
}
```

Per ulteriori informazioni, consulta [Modificare le regole del filtro Traffic Mirror](#) nella AWS Traffic Mirroring Guide.

- Per API i dettagli, consulta [DeleteTrafficMirrorFilterRule AWS CLI](#) Command Reference.

## delete-traffic-mirror-filter

Il seguente esempio di codice mostra come utilizzare `delete-traffic-mirror-filter`.

### AWS CLI

Per eliminare un filtro Traffic Mirror

L'`delete-traffic-mirror-filter`esempio seguente elimina il filtro Traffic Mirror specificato.

```
aws ec2 delete-traffic-mirror-filter \  
  --traffic-mirror-filter-id tmf-0be0b25fcdEXAMPLE
```

Output:

```
{  
  "TrafficMirrorFilterId": "tmf-0be0b25fcdEXAMPLE"
```

```
}
```

Per ulteriori informazioni, consulta [Eliminare un filtro Traffic Mirror](#) nella AWS Traffic Mirroring Guide.

- Per API i dettagli, consulta [DeleteTrafficMirrorFilter AWS CLI](#) Command Reference.

## **delete-traffic-mirror-session**

Il seguente esempio di codice mostra come utilizzare `delete-traffic-mirror-session`.

### AWS CLI

Per eliminare una sessione di Traffic Mirror

L'`delete-traffic-mirror-session` esempio seguente elimina la sessione mirror del traffico specificata.

```
aws ec2 delete-traffic-mirror-session \
  --traffic-mirror-session-id tms-0af3141ce5EXAMPLE
```

Output:

```
{
  "TrafficMirrorSessionId": "tms-0af3141ce5EXAMPLE"
}
```

Per ulteriori informazioni, consulta [Eliminare una sessione di Traffic Mirror nella Traffic Mirroring](#) AWS Guide.

- Per API i dettagli, consulta [DeleteTrafficMirrorSession AWS CLI](#) Command Reference.

## **delete-traffic-mirror-target**

Il seguente esempio di codice mostra come utilizzare `delete-traffic-mirror-target`.

### AWS CLI

Per eliminare un target del Traffic Mirror

L'`delete-traffic-mirror-target` esempio seguente elimina il target del Traffic Mirror specificato.

```
aws ec2 delete-traffic-mirror-target \  
  --traffic-mirror-target-id tmt-060f48ce9EXAMPLE
```

Output:

```
{  
  "TrafficMirrorTargetId": "tmt-060f48ce9EXAMPLE"  
}
```

Per ulteriori informazioni, consulta [Eliminare un Traffic Mirror Target nella AWS Traffic Mirroring Guide](#).

- Per API i dettagli, consulta [DeleteTrafficMirrorTarget AWS CLI Command Reference](#).

## delete-transit-gateway-connect-peer

Il seguente esempio di codice mostra come utilizzare `delete-transit-gateway-connect-peer`.

AWS CLI

Per eliminare un peer Transit Gateway Connect

L'esempio seguente elimina il peer Connect specificato.

```
aws ec2 delete-transit-gateway-connect-peer \  
  --transit-gateway-connect-peer-id tgw-connect-peer-0666adbac4EXAMPLE
```

Output:

```
{  
  "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"
      }
    ]
  }
}
}
}

```

Per ulteriori informazioni, consulta [gli allegati Transit gateway Connect e i peer Transit Gateway Connect](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [DeleteTransitGatewayConnectPeer](#) in AWS CLI Command Reference.

## delete-transit-gateway-connect

Il seguente esempio di codice mostra come utilizzare `delete-transit-gateway-connect`.

### AWS CLI

Per eliminare un gateway di transito (allegato Connect)

L'esempio seguente elimina l'allegato Connect specificato.

```
aws ec2 delete-transit-gateway-connect \
  --transit-gateway-attachment-id tgw-attach-037012e5dcEXAMPLE
```

Output:

```
{
```

```

    "TransitGatewayConnect": {
      "TransitGatewayAttachmentId": "tgw-attach-037012e5dcEXAMPLE",
      "TransportTransitGatewayAttachmentId": "tgw-attach-0a89069f57EXAMPLE",
      "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
      "State": "deleting",
      "CreationTime": "2021-03-09T19:59:17+00:00",
      "Options": {
        "Protocol": "gre"
      }
    }
  }
}

```

Per ulteriori informazioni, consulta [gli allegati Transit gateway Connect e i peer Transit Gateway Connect](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [Delete TransitGatewayConnect](#) in AWS CLI Command Reference.

## delete-transit-gateway-multicast-domain

Il seguente esempio di codice mostra come utilizzare `delete-transit-gateway-multicast-domain`.

### AWS CLI

Per eliminare un dominio multicast del gateway di transito

L'`delete-transit-gateway-multicast-domain` esempio seguente elimina il dominio multicast specificato.

```

aws ec2 delete-transit-gateway-multicast-domain \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE

```

Output:

```

{
  "TransitGatewayMulticastDomain": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-02bb79002bEXAMPLE",
    "TransitGatewayId": "tgw-0d88d2d0d5EXAMPLE",
    "State": "deleting",
    "CreationTime": "2019-11-20T22:02:03.000Z"
  }
}

```

Per ulteriori informazioni, vedere [Managing multicast Domains](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [Delete Transit Gateway Multicast Domain](#) in AWS CLI Command Reference.

## delete-transit-gateway-peering-attachment

Il seguente esempio di codice mostra come utilizzare `delete-transit-gateway-peering-attachment`.

### AWS CLI

Per eliminare un allegato di peering del gateway di transito

L'`delete-transit-gateway-peering-attachment` seguente elimina l'allegato di peering del gateway di transito specificato.

```
aws ec2 delete-transit-gateway-peering-attachment \
  --transit-gateway-attachment-id tgw-attach-4455667788aabbccd
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Transit Gateway Peering Attachments](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [DeleteTransitGatewayPeeringAttachment](#) in AWS CLI Command Reference.

## delete-transit-gateway-policy-table

Il seguente esempio di codice mostra come utilizzare `delete-transit-gateway-policy-table`.

### AWS CLI

Per eliminare una tabella di policy del gateway di transito

L'esempio seguente elimina la tabella delle politiche del gateway di transito specificata.

```
aws ec2 delete-transit-gateway-policy-table \
  --transit-gateway-policy-table-id tgw-ptb-0a16f134b78668a81
```

Output:

```
{
  "TransitGatewayPolicyTables": [
    {
      "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
      "TransitGatewayId": "tgw-067f8505c18f0bd6e",
      "State": "deleting",
      "CreationTime": "2023-11-28T16:36:43+00:00",
      "Tags": []
    }
  ]
}
```

Per ulteriori informazioni, consulta le [tabelle delle policy del Transit Gateway](#) nella Transit Gateway User Guide.

- Per API i dettagli, vedere [DeleteTransitGatewayPolicyTable](#) in AWS CLI Command Reference.

## delete-transit-gateway-prefix-list-reference

Il seguente esempio di codice mostra come utilizzare `delete-transit-gateway-prefix-list-reference`.

## AWS CLI

Per eliminare un riferimento all'elenco di prefissi

L'`delete-transit-gateway-prefix-list-reference` esempio seguente elimina il riferimento all'elenco di prefissi specificato.

```
aws ec2 delete-transit-gateway-prefix-list-reference \  
  --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \  
  --prefix-list-id pl-1111112222222333
```

Output:

```
{  
  "TransitGatewayPrefixListReference": {  
    "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",  
    "PrefixListId": "pl-1111112222222333",  
    "PrefixListOwnerId": "123456789012",  
    "State": "deleting",  
    "Blackhole": false,  
    "TransitGatewayAttachment": {  
      "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",  
      "ResourceType": "vpc",  
      "ResourceId": "vpc-112233445566aabbcc"  
    }  
  }  
}
```

Per ulteriori informazioni, vedete i [riferimenti all'elenco dei prefissi](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [DeleteTransitGatewayPrefixListReference](#) in AWS CLI Command Reference.

## `delete-transit-gateway-route-table`

Il seguente esempio di codice mostra come utilizzare `delete-transit-gateway-route-table`.

## AWS CLI

Per eliminare una tabella di routing del gateway di transito

L'`delete-transit-gateway-route-table` esempio seguente elimina la tabella di routing del gateway di transito specificata.

```
aws ec2 delete-transit-gateway-route-table \  
--transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE
```

Output:

```
{  
  "TransitGatewayRouteTable": {  
    "TransitGatewayRouteTableId": "tgw-rtb-0b6f6aaa01EXAMPLE",  
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",  
    "State": "deleting",  
    "DefaultAssociationRouteTable": false,  
    "DefaultPropagationRouteTable": false,  
    "CreationTime": "2019-07-17T20:27:26.000Z"  
  }  
}
```

Per ulteriori informazioni, consulta la [tabella di routing di un gateway di transito](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [DeleteTransitGatewayRouteTable](#) in AWS CLI Command Reference.

## delete-transit-gateway-route

Il seguente esempio di codice mostra come utilizzare `delete-transit-gateway-route`.

### AWS CLI

Per eliminare un CIDR blocco da una tabella di rotte

L'`delete-transit-gateway-route` esempio seguente elimina il CIDR blocco dalla tabella di routing del gateway di transito specificata.

```
aws ec2 delete-transit-gateway-route \  
--transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE \  
--destination-cidr-block 10.0.2.0/24
```

Output:

```
{  
  "Route": {  
    "DestinationCidrBlock": "10.0.2.0/24",  
    "TransitGatewayAttachments": [  

```

```

    {
      "ResourceId": "vpc-0065acced4EXAMPLE",
      "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
      "ResourceType": "vpc"
    }
  ],
  "Type": "static",
  "State": "deleted"
}
}

```

Per ulteriori informazioni, consulta [Eliminare una route statica](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [DeleteTransitGatewayRoute](#) in AWS CLI Command Reference.

## delete-transit-gateway-vpc-attachment

Il seguente esempio di codice mostra come utilizzare `delete-transit-gateway-vpc-attachment`.

### AWS CLI

Per eliminare un VPC allegato del gateway di transito

L'`delete-transit-gateway-vpc-attachment` seguente elimina l'VPC allegato specificato.

```

aws ec2 delete-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-0d2c54bdbEXAMPLE

```

Output:

```

{
  "TransitGatewayVpcAttachment": {
    "TransitGatewayAttachmentId": "tgw-attach-0d2c54bdb3EXAMPLE",
    "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
    "VpcId": "vpc-0065acced4f61c651",
    "VpcOwnerId": "111122223333",
    "State": "deleting",
    "CreationTime": "2019-07-17T16:04:27.000Z"
  }
}

```

Per ulteriori informazioni, vedete [Eliminare un VPC allegato](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [DeleteTransitGatewayVpcAttachment](#) in AWS CLI Command Reference.

## delete-transit-gateway

Il seguente esempio di codice mostra come utilizzare `delete-transit-gateway`.

### AWS CLI

Per eliminare un gateway di transito

L'`delete-transit-gateway` esempio seguente elimina il gateway di transito specificato.

```
aws ec2 delete-transit-gateway \  
  --transit-gateway-id tgw-01f04542b2EXAMPLE
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, vedere [Eliminare un gateway di transito](#) nella Guida ai gateway di transito.



- Per API i dettagli, vedere [Delete Transit Gateway](#) in AWS CLI Command Reference.

## delete-verified-access-endpoint

Il seguente esempio di codice mostra come utilizzare `delete-verified-access-endpoint`.

### AWS CLI

Per eliminare un endpoint di accesso verificato

L'`delete-verified-access-endpoint` seguente elimina l'endpoint Verified Access specificato.

```
aws ec2 delete-verified-access-endpoint \  
--verified-access-endpoint-id vae-066fac616d4d546f2
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta gli [endpoint di accesso verificato nella Guida](#) per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [DeleteVerifiedAccessEndpoint AWS CLI Command Reference](#).

## delete-verified-access-group

Il seguente esempio di codice mostra come utilizzare `delete-verified-access-group`.

### AWS CLI

Per eliminare un gruppo con accesso verificato

L'esempio seguente elimina il gruppo di accesso verificato specificato.

```
aws ec2 delete-verified-access-group \  
--verified-access-group-id vagr-0dbe967baf14b7235
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Gruppi di accesso verificato](#) nella Guida per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [DeleteVerifiedAccessGroup AWS CLI Command Reference](#).

## delete-verified-access-instance

Il seguente esempio di codice mostra come utilizzare `delete-verified-access-instance`.

### AWS CLI

Per eliminare un'istanza di accesso verificato

L'esempio seguente elimina l'istanza di accesso verificato specificata.

```
aws ec2 delete-verified-access-instance \  
  --verified-access-instance-id vai-0ce000c0b7643abea
```

Output:

```
{  
  "VerifiedAccessInstance": {  
    "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",  
    "Description": "Testing Verified Access",  
    "VerifiedAccessTrustProviders": [],  
    "CreationTime": "2023-08-25T18:27:56",  
    "LastUpdatedTime": "2023-08-26T01:00:18"  
  }  
}
```

Per ulteriori informazioni, consulta le [istanze di accesso verificato nella Guida](#) per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [DeleteVerifiedAccessInstance AWS CLI Command Reference](#).

## delete-verified-access-trust-provider

Il seguente esempio di codice mostra come utilizzare `delete-verified-access-trust-provider`.

### AWS CLI

Per eliminare un provider fiduciario di accesso verificato

L'delete-verified-access-trust-provider esempio seguente elimina il provider fiduciario Verified Access specificato.

```
aws ec2 delete-verified-access-trust-provider \
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Trust providers for Verified Access](#) nella AWS Verified Access User Guide.

- Per API i dettagli, consulta [DeleteVerifiedAccessTrustProvider AWS CLI Command Reference](#).

## delete-volume

Il seguente esempio di codice mostra come utilizzare delete-volume.

### AWS CLI

Per eliminare un volume

Questo comando di esempio elimina un volume disponibile con l'ID del volume divo1-049df61146c4d7901. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-volume --volume-id vol-049df61146c4d7901
```

- Per API i dettagli, vedere [DeleteVolume](#) in AWS CLI Command Reference.

## delete-vpc-endpoint-connection-notifications

Il seguente esempio di codice mostra come utilizzare `delete-vpc-endpoint-connection-notifications`.

### AWS CLI

Per eliminare una notifica di connessione all'endpoint

Questo esempio elimina la notifica di connessione all'endpoint specificata.

Comando:

```
aws ec2 delete-vpc-endpoint-connection-notifications --connection-notification-ids vpce-nfn-008776de7e03f5abc
```

Output:

```
{
  "Unsuccessful": []
}
```

- Per API i dettagli, vedere [DeleteVpcEndpointConnectionNotifications](#) in AWS CLI Command Reference.

## delete-vpc-endpoint-service-configurations

Il seguente esempio di codice mostra come utilizzare `delete-vpc-endpoint-service-configurations`.

### AWS CLI

Per eliminare una configurazione del servizio endpoint

Questo esempio elimina la configurazione del servizio endpoint specificata.

Comando:

```
aws ec2 delete-vpc-endpoint-service-configurations --service-ids vpce-svc-03d5ebb7d9579a2b3
```

**Output:**

```
{
  "Unsuccessful": []
}
```

- Per API i dettagli, vedere [DeleteVpcEndpointServiceConfigurations](#) in AWS CLI Command Reference.

**delete-vpc-endpoints**

Il seguente esempio di codice mostra come utilizzare `delete-vpc-endpoints`.

**AWS CLI**

Per eliminare un endpoint

Questo esempio elimina gli endpoint `vpce-aa22bb33` e `vpce-1a2b3c4d`. Se il comando ha esito positivo o negativo in parte, viene restituito un elenco di elementi non riusciti. Se il comando ha esito positivo, l'elenco restituito è vuoto.

Comando:

```
aws ec2 delete-vpc-endpoints --vpc-endpoint-ids vpce-aa22bb33 vpce-1a2b3c4d
```

**Output:**

```
{
  "Unsuccessful": []
}
```

- Per API i dettagli, vedere [DeleteVpcEndpoints](#) in AWS CLI Command Reference.

**delete-vpc-peering-connection**

Il seguente esempio di codice mostra come utilizzare `delete-vpc-peering-connection`.

**AWS CLI**

Per eliminare una connessione VPC peering

Questo esempio elimina la connessione VPC peering specificata.

Comando:

```
aws ec2 delete-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

Output:

```
{
  "Return": true
}
```

- Per API i dettagli, vedere [DeleteVpcPeeringConnection](#) in AWS CLI Command Reference.

## **delete-vpc**

Il seguente esempio di codice mostra come utilizzare `delete-vpc`.

AWS CLI

Per eliminare un VPC

Questo esempio elimina il valore specificato VPC. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-vpc --vpc-id vpc-a01106c2
```

- Per API i dettagli, vedere [DeleteVpc](#) in AWS CLI Command Reference.

## **delete-vpn-connection-route**

Il seguente esempio di codice mostra come utilizzare `delete-vpn-connection-route`.

AWS CLI

Per eliminare una route statica da una VPN connessione

Questo esempio elimina la route statica specificata dalla VPN connessione specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-cidr-block 11.12.0.0/16
```

- Per API i dettagli, vedere [DeleteVpnConnectionRoute](#) in AWS CLI Command Reference.

## delete-vpn-connection

Il seguente esempio di codice mostra come utilizzare `delete-vpn-connection`.

AWS CLI

Per eliminare una VPN connessione

Questo esempio elimina la VPN connessione specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-vpn-connection --vpn-connection-id vpn-40f41529
```

- Per API i dettagli, vedere [DeleteVpnConnection](#) in AWS CLI Command Reference.

## delete-vpn-gateway

Il seguente esempio di codice mostra come utilizzare `delete-vpn-gateway`.

AWS CLI

Per eliminare un gateway privato virtuale

Questo esempio elimina il gateway privato virtuale specificato. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 delete-vpn-gateway --vpn-gateway-id vgw-9a4cacf3
```

- Per API i dettagli, vedere [DeleteVpnGateway](#) in AWS CLI Command Reference.



## deprovision-byoip-cidr

Il seguente esempio di codice mostra come utilizzare `deprovision-byoip-cidr`.

### AWS CLI

Per rimuovere un intervallo di indirizzi IP dall'uso

L'esempio seguente rimuove l'intervallo di indirizzi specificato dall'uso con AWS.

```
aws ec2 deprovision-byoip-cidr \  
  --cidr 203.0.113.25/24
```

Output:

```
{  
  "ByoipCidr": {  
    "Cidr": "203.0.113.25/24",  
    "State": "pending-deprovision"  
  }  
}
```

- Per API i dettagli, vedere [DeprovisionByoipCidr](#) in AWS CLI Command Reference.

## deprovision-ipam-pool-cidr

Il seguente esempio di codice mostra come utilizzare `deprovision-ipam-pool-cidr`.

### AWS CLI

Effettuare il deprovisioning di un pool IPAM CIDR

L'`deprovision-ipam-pool-cidr` esempio seguente esegue il deprovisioning di un CIDR provisioning in un pool. 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
```

Output:

```
{  
  "IpamPoolCidr": {  
    "Cidr": "11.0.0.0/16",  
    "State": "pending-deprovision"  
  }  
}
```

Per ulteriori informazioni, consulta [Deprovision pool CIDRs](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [DeprovisionIpamPoolCidr AWS CLI Command Reference](#).

## deregister-image

Il seguente esempio di codice mostra come utilizzare `deregister-image`.

### AWS CLI

Per annullare la registrazione di un AMI

Questo esempio annulla la registrazione di quanto specificato. AMI Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 deregister-image --image-id ami-4fa54026
```

- Per API i dettagli, vedere [DeregisterImage](#) in AWS CLI Command Reference.

## deregister-instance-event-notification-attributes

Il seguente esempio di codice mostra come utilizzare `deregister-instance-event-notification-attributes`.

## AWS CLI

Esempio 1: per rimuovere tutti i tag dalle notifiche degli eventi

L'operazione `deregister-instance-event-notification-attributes` seguente rimuove `IncludeAllTagsOfInstance=true`, che ha l'effetto di impostare `IncludeAllTagsOfInstance` su `false`.

```
aws ec2 deregister-instance-event-notification-attributes \  
--instance-tag-attribute IncludeAllTagsOfInstance=true
```

Output:

```
{  
  "InstanceTagAttribute": {  
    "InstanceTagKeys": [],  
    "IncludeAllTagsOfInstance": true  
  }  
}
```

Per ulteriori informazioni, consulta [Eventi pianificati per le tue istanze](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

Esempio 2: per rimuovere tag specifici dalle notifiche degli eventi

L'operazione `deregister-instance-event-notification-attributes` seguente rimuove il tag specificato dai tag inclusi nelle notifiche degli eventi. Per descrivere i tag rimanenti inclusi nelle notifiche degli eventi, usa `describe-instance-event-notification-attributes`.

```
aws ec2 deregister-instance-event-notification-attributes \  
--instance-tag-attribute InstanceTagKeys="tag-key2"
```

Output:

```
{  
  "InstanceTagAttribute": {  
    "InstanceTagKeys": [  
      "tag-key2"  
    ],  
    "IncludeAllTagsOfInstance": false  
  }  
}
```

```
}
```

Per ulteriori informazioni, consulta [Eventi pianificati per le tue istanze](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per API i dettagli, consulta Command [DeregisterInstanceEventNotificationAttributesReference](#) AWS CLI .

## **deregister-transit-gateway-multicast-group-members**

Il seguente esempio di codice mostra come utilizzare `deregister-transit-gateway-multicast-group-members`.

### AWS CLI

Per annullare la registrazione dei membri di un gruppo multicast

Questo esempio annulla la registrazione del membro del gruppo di interfaccia di rete specificato dal gruppo multicast del gateway di transito.

```
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
```

Output:

```
{  
  "DeregisteredMulticastGroupMembers": {  
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",  
    "RegisteredNetworkInterfaceIds": [  
      "eni-0e246d3269EXAMPLE"  
    ],  
    "GroupIpAddress": "224.0.1.0"  
  }  
}
```

Per ulteriori informazioni, consulta Annullare la [registrazione dei membri da un gruppo multicast](#) nella Guida per l'utente di Transit Gateways.AWS

- Per i API dettagli, consulta [DeregisterTransitGatewayMulticastGroupMembersCommand](#) Reference.AWS CLI

## deregister-transit-gateway-multicast-group-source

Il seguente esempio di codice mostra come utilizzare `deregister-transit-gateway-multicast-group-source`.

### AWS CLI

Per annullare la registrazione di una sorgente dal gruppo multicast del gateway di transito

Questo esempio annulla la registrazione dell'origine del gruppo di interfacce di rete specificato dal gruppo multicast.

```
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
```

Output:

```
{
  "DeregisteredMulticastGroupSources": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
    "DeregisteredNetworkInterfaceIds": [
      "eni-07f290fc3c090cbae"
    ],
    "GroupIpAddress": "224.0.1.0"
  }
}
```

Per ulteriori informazioni, vedete Annullare la [registrazione delle fonti da un gruppo multicast](#) nella Guida per l'utente di Transit Gateways.AWS

- Per i API dettagli, [DeregisterTransitGatewayMulticastGroupSource](#)consultate Command Reference.AWS CLI

## describe-account-attributes

Il seguente esempio di codice mostra come utilizzare `describe-account-attributes`.

### AWS CLI

Per descrivere tutti gli attributi del tuo AWS account

Questo esempio descrive gli attributi del tuo AWS account.

Comando:

```
aws ec2 describe-account-attributes
```

Output:

```
{
  "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"
      }
    ]
  }
]
```

Per descrivere un singolo attributo del tuo AWS account

Questo esempio descrive l'`supported-platforms` attributo del tuo AWS account.

Comando:

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
```

Output:

```
{
  "AccountAttributes": [
    {
      "AttributeName": "supported-platforms",
      "AttributeValues": [
        {
          "AttributeValue": "EC2"
        },
        {
          "AttributeValue": "VPC"
        }
      ]
    }
  ]
}
```

```
    }  
  ]  
}
```

- Per API i dettagli, consulta [DescribeAccountAttributes AWS CLI Command Reference](#).

## describe-address-transfers

Il seguente esempio di codice mostra come utilizzare `describe-address-transfers`.

### AWS CLI

Per descrivere un trasferimento di indirizzi IP elastico

L'esempio seguente descrive il trasferimento di indirizzi IP elastici per l'indirizzo IP elastico specificato.

```
aws ec2 describe-address-transfers \  
  --allocation-ids eipalloc-09ad461b0d03f6aaf
```

Output:

```
{  
  "AddressTransfers": [  
    {  
      "PublicIp": "100.21.184.216",  
      "AllocationId": "eipalloc-09ad461b0d03f6aaf",  
      "TransferAccountId": "123456789012",  
      "TransferOfferExpirationTimestamp": "2023-02-22T22:51:01.000Z",  
      "AddressTransferStatus": "pending"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Transfer Elastic IP address](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [DescribeAddressTransfers AWS CLI Command Reference](#).

## describe-addresses-attribute

Il seguente esempio di codice mostra come utilizzare `describe-addresses-attribute`.



## AWS CLI

Per visualizzare gli attributi del nome di dominio associato a un indirizzo IP elastico

`describe-addresses-attribute` Gli esempi seguenti restituiscono gli attributi del nome di dominio associato all'indirizzo IP elastico.

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
```

Output:

```
{  
  "Addresses": [  
    {  
      "PublicIp": "192.0.2.0",  
      "AllocationId": "eipalloc-abcdef01234567890",  
      "PtrRecord": "example.com."  
    }  
  ]  
}
```

Per visualizzare gli attributi di un indirizzo IP elastico, è necessario prima aver associato un nome di dominio all'indirizzo IP elastico. Per ulteriori informazioni, consulta [Use reverse DNS for email application](#) nella Amazon EC2 User Guide o [modify-address-attribute](#) nel AWS CLI Command Reference.

- Per API i dettagli, consulta [DescribeAddressesAttribute AWS CLI Command Reference](#).

## `describe-addresses`

Il seguente esempio di codice mostra come utilizzare `describe-addresses`.

## AWS CLI

Esempio 1: per recuperare i dettagli di tutti gli indirizzi IP elastici

Nell'esempio di `describe addresses` seguente vengono visualizzati tutti i dettagli relativi agli indirizzi IP elastici.

```
aws ec2 describe-addresses
```

Output:

```
{
  "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"
    }
  ]
}
```

Esempio 2: per recuperare i dettagli degli indirizzi IP elastici per EC2 - VPC

L'`describe-addresses` seguente mostra i dettagli sugli indirizzi IP elastici da utilizzare con le istanze in a. VPC

```
aws ec2 describe-addresses \
  --filters "Name=domain,Values=vpc"
```

Output:

```
{
  "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"
    }
  ]
}
```

Esempio 3: per recuperare i dettagli di un indirizzo IP elastico specificato dall'ID di allocazione

L'`describe-addresses` seguente visualizza i dettagli sull'indirizzo IP elastico con l'ID di allocazione specificato, associato a un'istanza in EC2 - VPC

```
aws ec2 describe-addresses \
  --allocation-ids eipalloc-282d9641
```

Output:

```
{
  "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"
    }
  ]
}
```

Esempio 4: per recuperare i dettagli su un indirizzo IP elastico specificato dal relativo indirizzo IP VPC privato

L'output del comando `aws ec2 describe-addresses` seguente mostra i dettagli sull'indirizzo IP elastico associato a un particolare indirizzo IP privato in EC2 -VPC.

```
aws ec2 describe-addresses \
  --filters "Name=private-ip-address,Values=10.251.50.12"
```

Esempio 5: Per recuperare i dettagli sugli indirizzi IP elastici in EC2 -Classic

L'output del comando `aws ec2 describe-addresses` seguente mostra i dettagli sugli indirizzi IP elastici da utilizzare in EC2 -Classic.

```
aws ec2 describe-addresses \
  --filters "Name=domain,Values=standard"
```

Output:

```
{
  "Addresses": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "PublicIp": "203.0.110.25",
      "PublicIpv4Pool": "amazon",
      "Domain": "standard"
    }
  ]
}
```

Esempio 6: per recuperare i dettagli di un indirizzo IP elastico specificato dall'indirizzo IP pubblico

L'output del comando `aws ec2 describe-addresses` seguente visualizza i dettagli sull'indirizzo IP elastico con il valore `203.0.110.25`, che è associato a un'istanza in EC2 -Classic.

```
aws ec2 describe-addresses \
  --public-ips 203.0.110.25
```

Output:

```
{
```

```
"Addresses": [  
  {  
    "InstanceId": "i-1234567890abcdef0",  
    "PublicIp": "203.0.110.25",  
    "PublicIpv4Pool": "amazon",  
    "Domain": "standard"  
  }  
]  
}
```

- Per API i dettagli, vedere [DescribeAddresses](#) in AWS CLI Command Reference.

## describe-aggregate-id-format

Il seguente esempio di codice mostra come utilizzare `describe-aggregate-id-format`.

### AWS CLI

Per descrivere le impostazioni del formato ID più lungo per tutti i tipi di risorse in una regione

L'`describe-aggregate-id-format` esempio seguente descrive lo stato complessivo del formato ID lungo per la regione corrente. Il `Deadline` valore indica che le scadenze per il passaggio permanente di queste risorse dal formato ID breve al formato ID lungo sono scadute. Il `UseLongIdsAggregated` valore indica che tutti IAM gli utenti e i IAM ruoli sono configurati per utilizzare il formato ID lungo per tutti i tipi di risorse.

```
aws ec2 describe-aggregate-id-format
```

Output:

```
{  
  "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
      },
      ...
    ]
  }
}
```

- Per API i dettagli, vedere [DescribeAggregateIdFormat](#) in AWS CLI Command Reference.

## describe-availability-zones

Il seguente esempio di codice mostra come utilizzare `describe-availability-zones`.

### AWS CLI

Per descrivere le zone di disponibilità

In questo esempio di `describe-availability-zones` vengono mostrati i dettagli della zona di disponibilità disponibili a te. La risposta include le zone di disponibilità solo per la regione attuale. In questo esempio si utilizza la regione predefinita del profilo `us-west-2` (Oregon).

```
aws ec2 describe-availability-zones
```

Output:

```
{
  "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"
}
]
```

- Per API i dettagli, vedere [DescribeAvailabilityZones](#) in AWS CLI Command Reference.

## describe-aws-network-performance-metric-subscription

Il seguente esempio di codice mostra come utilizzare `describe-aws-network-performance-metric-subscription`.

### AWS CLI

Per descrivere i tuoi abbonamenti metrici

L'`describe-aws-network-performance-metric-subscription` seguente descrive i tuoi abbonamenti metrici.

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

Output:

```
{
  "Subscriptions": [
    {
      "Source": "us-east-1",
      "Destination": "eu-west-1",
      "Metric": "aggregate-latency",
      "Statistic": "p50",
      "Period": "five-minutes"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Manage subscriptions](#) nella Infrastructure Performance User Guide.

- Per API i dettagli, vedere [DescribeAwsNetworkPerformanceMetricSubscription](#) in AWS CLI Command Reference.

## describe-aws-network-performance-metric-subscriptions

Il seguente esempio di codice mostra come utilizzare `describe-aws-network-performance-metric-subscriptions`.

### AWS CLI

Per descrivere i tuoi abbonamenti metrici



L'`describe-aws-network-performance-metric-subscriptions` seguente descrive i tuoi abbonamenti metrici.

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

Output:

```
{
  "Subscriptions": [
    {
      "Source": "us-east-1",
      "Destination": "eu-west-1",
      "Metric": "aggregate-latency",
      "Statistic": "p50",
      "Period": "five-minutes"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Manage subscriptions](#) nella Infrastructure Performance User Guide.

- Per API i dettagli, vedere [DescribeAwsNetworkPerformanceMetricSubscriptions](#) in AWS CLI Command Reference.

## **describe-bundle-tasks**

Il seguente esempio di codice mostra come utilizzare `describe-bundle-tasks`.

AWS CLI

Per descrivere le attività del pacchetto

Questo esempio descrive tutte le attività del pacchetto.

Comando:

```
aws ec2 describe-bundle-tasks
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, consulta [DescribeBundleTasks AWS CLI](#) Command Reference.

## describe-byoip-cidrs

Il seguente esempio di codice mostra come utilizzare `describe-byoip-cidrs`.

### AWS CLI

Per descrivere gli intervalli di indirizzi forniti

L'esempio seguente mostra i dettagli sugli intervalli di IPv4 indirizzi pubblici di cui hai predisposto l'uso. AWS

```
aws ec2 describe-byoip-cidrs
```

Output:

```
{
  "ByoipCidrs": [
    {
      "Cidr": "203.0.113.25/24",
      "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
    }
  ]
}
```

```
        "State": "provisioned"
      }
    ]
  }
```

- Per API i dettagli, vedere [DescribeByoipCidrs](#) in AWS CLI Command Reference.

## describe-capacity-reservation-fleets

Il seguente esempio di codice mostra come utilizzare `describe-capacity-reservation-fleets`.

### AWS CLI

Per visualizzare una flotta di prenotazioni di capacità

L'esempio seguente elenca le informazioni sulla configurazione e sulla capacità per la flotta di prenotazioni di capacità specificata. Elenca inoltre i dettagli sulle singole prenotazioni di capacità che si trovano all'interno della flotta. :

```
aws ec2 describe-capacity-reservation-fleets \
  --capacity-reservation-fleet-ids crf-abcdef01234567890
```

Output:

```
{
  "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"
  }
]
```

Per ulteriori informazioni sulle flotte di prenotazione della capacità, consulta [Flotte di prenotazione della capacità nella Guida](#) per gli EC2 utenti di Amazon.

- Per API i dettagli, consulta AWS CLI Command [DescribeCapacityReservationFleets](#) Reference.

## describe-capacity-reservations

Il seguente esempio di codice mostra come utilizzare `describe-capacity-reservations`.

### AWS CLI

Esempio 1: Per descrivere una o più prenotazioni di capacità

L'esempio seguente mostra i dettagli su tutte le prenotazioni di capacità nella AWS regione corrente.

```
aws ec2 describe-capacity-reservations
```

Output:

```
{
  "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"
  }
]
}

```

Esempio 2: Per descrivere una o più delle tue prenotazioni di capacità

L'`describe-capacity-reservations` seguente mostra i dettagli sulla prenotazione di capacità specificata.

```

aws ec2 describe-capacity-reservations \
  --capacity-reservation-ids cr-1234abcd56EXAMPLE

```

Output:

```

{

```

```
"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"  
  }  
]  
}
```

Per ulteriori informazioni, consulta [Visualizzazione di una prenotazione di capacità](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per API i dettagli, consulta AWS CLI Command [DescribeCapacityReservations](#) Reference.

## describe-carrier-gateways

Il seguente esempio di codice mostra come utilizzare `describe-carrier-gateways`.

AWS CLI

Per descrivere tutti i carrier gateway

L'`describe-carrier-gateways` esempio seguente elenca tutti i gateway del tuo operatore.

```
aws ec2 describe-carrier-gateways
```

Output:

```
{  
  "CarrierGateways": [  
    {
```

```
    "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
    "VpcId": "vpc-0c529aEXAMPLE",
    "State": "available",
    "OwnerId": "123456789012",
    "Tags": [
      {
        "Key": "example",
        "Value": "tag"
      }
    ]
  }
]
```

Per ulteriori informazioni, consulta Carrier gateways < [https://docs.aws.amazon.com/vpc/latest/userguide/Carrier\\_gateway.html](https://docs.aws.amazon.com/vpc/latest/userguide/Carrier_gateway.html) > nella Amazon Virtual Private Cloud User Guide.

- Per i API AWS CLI dettagli, [DescribeCarrierGateways](#) consulta Command Reference.

## describe-classic-link-instances

Il seguente esempio di codice mostra come utilizzare `describe-classic-link-instances`.

### AWS CLI

Per descrivere le istanze EC2 -Classic collegate

Questo esempio elenca tutte le istanze EC2 -Classic collegate.

Comando:

```
aws ec2 describe-classic-link-instances
```

Output:

```
{
  "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"
        }
    ]
}
]
}

```

Questo esempio elenca tutte le istanze EC2 -Classic collegate e filtra la risposta per includere solo le istanze collegate a vpc-88888888. VPC

Comando:

```
aws ec2 describe-classic-link-instances --filter "Name=vpc-id,Values=vpc-88888888"
```

Output:

```

{
    "Instances": [
        {
            "InstanceId": "i-1234567890abcdef0",

```



```

        "VpcId": "vpc-88888888",
        "Groups": [
            {
                "GroupId": "sg-11122233"
            }
        ],
        "Tags": [
            {
                "Value": "ClassicInstance",
                "Key": "Name"
            }
        ]
    }
]
}

```

- Per i API dettagli, consulta Command Reference. [DescribeClassicLinkInstances](#) AWS CLI

## describe-client-vpn-authorization-rules

Il seguente esempio di codice mostra come utilizzare `describe-client-vpn-authorization-rules`.

### AWS CLI

Per descrivere le regole di autorizzazione per un VPN endpoint Client

L'esempio seguente mostra i dettagli sulle regole di autorizzazione per l'VPN endpoint Client specificato.

```

aws ec2 describe-client-vpn-authorization-rules \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde

```

Output:

```

{
  "AuthorizationRules": [
    {
      "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
      "GroupId": "",
      "AccessAll": true,
    }
  ]
}

```

```

        "DestinationCidr": "0.0.0.0/0",
        "Status": {
            "Code": "active"
        }
    }
]
}

```

Per ulteriori informazioni, vedere [Regole di autorizzazione](#) nella AWS Client VPN Administrator Guide.

- Per API i dettagli, vedere [DescribeClientVpnAuthorizationRules](#) in AWS CLI Command Reference.

## describe-client-vpn-connections

Il seguente esempio di codice mostra come utilizzare `describe-client-vpn-connections`.

### AWS CLI

Per descrivere le connessioni a un VPN endpoint Client

L'esempio seguente mostra i dettagli sulle connessioni del client all'VPN endpoint Client specificato.

```

aws ec2 describe-client-vpn-connections \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde

```

Output:

```

{
  "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": "-"
  }
]
}

```

Per ulteriori informazioni, vedere [Client Connections nella Client VPN Administrator Guide](#).AWS

- Per API i dettagli, vedere [DescribeClientVpnConnections](#) in AWS CLI Command Reference.

## describe-client-vpn-endpoints

Il seguente esempio di codice mostra come utilizzare `describe-client-vpn-endpoints`.

### AWS CLI

Per descrivere gli VPN endpoint del client

L'esempio seguente mostra i dettagli su tutti gli VPN endpoint Client.

```
aws ec2 describe-client-vpn-endpoints
```

## Output:

```
{
  "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-aabbcc112233445566"
        ],
        "VpcId": "vpc-a87f92c1",
        "SelfServicePortalUrl": "https://self-service.clientvpn.amazonaws.com/
endpoints/cvpn-endpoint-123456789123abcde",
        "ClientConnectOptions": {
            "Enabled": false
        }
    }
]
}

```

Per ulteriori informazioni, consulta [Client VPN Endpoints](#) nella Client VPN Administrator AWS Guide.

- Per API i dettagli, vedere [DescribeClientVpnEndpoints](#) in AWS CLI Command Reference.

## describe-client-vpn-routes

Il seguente esempio di codice mostra come utilizzare `describe-client-vpn-routes`.

### AWS CLI

Per descrivere i percorsi per un VPN endpoint Client

L'esempio seguente mostra i dettagli sui percorsi per l'VPN endpoint Client specificato.

```

aws ec2 describe-client-vpn-routes \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde

```

Output:

```

{
  "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"
        }
    }
]
}

```

Per ulteriori informazioni, vedere [Routes](#) nella AWS Client VPN Administrator Guide.

- Per API i dettagli, vedere [DescribeClientVpnRoutes](#) in AWS CLI Command Reference.

## describe-client-vpn-target-networks

Il seguente esempio di codice mostra come utilizzare `describe-client-vpn-target-networks`.

### AWS CLI

Per descrivere le reti di destinazione per un VPN endpoint Client

L'esempio seguente mostra i dettagli sulle reti di destinazione per l'VPN endpoint Client specificato.

```

aws ec2 describe-client-vpn-target-networks \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde

```

Output:

```

{
  "ClientVpnTargetNetworks": [
    {
      "AssociationId": "cvpn-assoc-012e837060753dc3d",

```

```

    "VpcId": "vpc-1111122222233333",
    "TargetNetworkId": "subnet-0123456789abcabca",
    "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
    "Status": {
      "Code": "associating"
    },
    "SecurityGroups": [
      "sg-012345678910abcab"
    ]
  }
]
}

```

Per ulteriori informazioni, consulta [Target Networks](#) nella AWS Client VPN Administrator Guide.

- Per API i dettagli, vedere [DescribeClientVpnTargetNetworks](#) in AWS CLI Command Reference.

## describe-coip-pools

Il seguente esempio di codice mostra come utilizzare `describe-coip-pools`.

### AWS CLI

Per descrivere i pool di indirizzi IP di proprietà del cliente

L'`describe-coip-pools`esempio seguente descrive i pool di indirizzi IP di proprietà del cliente presenti nel tuo account. AWS

```
aws ec2 describe-coip-pools
```

Output:

```

{
  "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"
    }
  ]
}

```

```
]
}
```

Per ulteriori informazioni, consulta [Indirizzi IP di proprietà del cliente](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, vedere [DescribeCoipPools](#) in AWS CLI Command Reference.

## describe-conversion-tasks

Il seguente esempio di codice mostra come utilizzare `describe-conversion-tasks`.

### AWS CLI

Per visualizzare lo stato di un'operazione di conversione

Questo esempio restituisce lo stato di un'attività di conversione con l'ID `import-i-ffvko9js`.

Comando:

```
aws ec2 describe-conversion-tasks --conversion-task-ids import-i-ffvko9js
```

Output:

```
{
  "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"
}
]
}

```

- Per API i dettagli, vedere [DescribeConversionTasks](#) in AWS CLI Command Reference.

## describe-customer-gateways

Il seguente esempio di codice mostra come utilizzare `describe-customer-gateways`.

### AWS CLI

Per descrivere i gateway per i clienti

Questo esempio descrive i gateway per i clienti.

Comando:

```
aws ec2 describe-customer-gateways
```

Output:

```

{
  "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"
    }
]
}
```

Per descrivere un gateway specifico per i clienti

Questo esempio descrive il Customer Gateway specificato.

Comando:

```
aws ec2 describe-customer-gateways --customer-gateway-ids cgw-0e11f167
```

Output:

```
{
  "CustomerGateways": [
    {
      "CustomerGatewayId": "cgw-0e11f167",
      "IpAddress": "12.1.2.3",
      "State": "available",
      "Type": "ipsec.1",
      "BgpAsn": "65534"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeCustomerGateways](#) in AWS CLI Command Reference.

## describe-dhcp-options

Il seguente esempio di codice mostra come utilizzare `describe-dhcp-options`.

AWS CLI

Esempio 1: Per descrivere le DHCP opzioni

L'`describe-dhcp-options` esempio seguente recupera i dettagli sulle tue DHCP opzioni.

**aws ec2 describe-dhcp-options**

Output:

```
{
  "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"
}
]
}

```

Per ulteriori informazioni, vedere [Lavorare con i set di DHCP opzioni](#) nella Guida per l'AWS VPCutente.

Esempio 2: per descrivere le DHCP opzioni e filtrare l'output

L'esempio seguente descrive le DHCP opzioni disponibili e utilizza un filtro per restituire solo DHCP le opzioni disponibili `example.com` per il server dei nomi di dominio. L'esempio utilizza il `--query` parametro per visualizzare solo le informazioni di configurazione e l'ID nell'output.

```

aws ec2 describe-dhcp-options \
  --filters Name=key,Values=domain-name-servers Name=value,Values=example.com \
  --query "DhcpOptions[*].[DhcpConfigurations,DhcpOptionsId]"

```

Output:

```

[
  [
    [
      {
        "Key": "domain-name",
        "Values": [
          {
            "Value": "example.com"
          }
        ]
      },
      {
        "Key": "domain-name-servers",
        "Values": [
          {
            "Value": "172.16.16.16"
          }
        ]
      }
    ]
  ]
]

```

```
    ]
  },
  "dopt-001122334455667ab"
]
]
```

Per ulteriori informazioni, vedete [Lavorare con i set di DHCP opzioni](#) nella Guida AWS VPC per l'utente.

- Per API i dettagli, vedere [DescribeDhcpOptions](#) in AWS CLI Command Reference.

## describe-egress-only-internet-gateways

Il seguente esempio di codice mostra come utilizzare `describe-egress-only-internet-gateways`.

### AWS CLI

Per descrivere i gateway Internet utilizzati solo in uscita

Questo esempio descrive i gateway Internet solo in uscita.

Comando:

```
aws ec2 describe-egress-only-internet-gateways
```

Output:

```
{
  "EgressOnlyInternetGateways": [
    {
      "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
      "Attachments": [
        {
          "State": "attached",
          "VpcId": "vpc-0c62a468"
        }
      ]
    }
  ]
}
```

```
}
```

- Per i API dettagli, vedere [DescribeEgressOnlyInternetGateways](#) in Command Reference.AWS CLI

## describe-elastic-gpus

Il seguente esempio di codice mostra come utilizzare `describe-elastic-gpus`.

AWS CLI

Per descrivere un elastico GPU

Comando:

```
aws ec2 describe-elastic-gpus --elastic-gpu-ids egpu-12345678901234567890abcdefghijkl
```

- Per API i dettagli, vedere [DescribeElasticGpus](#) in AWS CLI Command Reference.

## describe-export-image-tasks

Il seguente esempio di codice mostra come utilizzare `describe-export-image-tasks`.

AWS CLI

Per monitorare un'operazione di esportazione di immagini

L'`describe-export-image-tasks`esempio seguente controlla lo stato dell'attività di esportazione dell'immagine specificata. Il file di immagine risultante in Amazon S3 è. `my-export-bucket/exports/export-ami-1234567890abcdef0.vmdk`

```
aws ec2 describe-export-image-tasks \--export-image-task-ids export-ami-1234567890abcdef0
```

Output per un'attività di esportazione di immagini in corso.

```
{  
  "ExportImageTasks": [  
    {  
      "ExportImageTaskId": "export-ami-1234567890abcdef0"    }  
  ]  
}
```

```

    "Progress": "21",
    "S3ExportLocation": {
      "S3Bucket": "my-export-bucket",
      "S3Prefix": "exports/"
    },
    "Status": "active",
    "StatusMessage": "updating"
  }
]
}

```

Output per un'operazione di esportazione di immagini completata.

```

{
  "ExportImageTasks": [
    {
      "ExportImageTaskId": "export-ami-1234567890abcdef0"
      "S3ExportLocation": {
        "S3Bucket": "my-export-bucket",
        "S3Prefix": "exports/"
      },
      "Status": "completed"
    }
  ]
}

```

Per ulteriori informazioni, vedere [Export a VM da un file AMI nella VM](#) Import/Export User Guide.

- Per i API dettagli, vedere [DescribeExportImageTasks](#) in Command Reference.AWS CLI

## describe-export-tasks

Il seguente esempio di codice mostra come utilizzare `describe-export-tasks`.

### AWS CLI

Per elencare i dettagli su un'attività di esportazione di istanze

Questo esempio descrive l'attività di esportazione con ID `export-i-fh8sjjsq`.

Comando:

```
aws ec2 describe-export-tasks --export-task-ids export-i-fh8sjjsq
```

**Output:**

```
{
  "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"
    }
  ]
}
```

- Per i API dettagli, vedere [DescribeExportTasks](#) in Command Reference.AWS CLI

**describe-fast-launch-images**

Il seguente esempio di codice mostra come utilizzare `describe-fast-launch-images`.

**AWS CLI**

Per descrivere i dettagli relativi a Windows AMIs configurati per un avvio più rapido

L'esempio seguente descrive i dettagli di ciascuno degli elementi dell'account configurati per un avvio più rapido, tra cui il tipo di risorsa, la configurazione dell'istantanea, i dettagli del modello di avvio, il numero massimo di avvii paralleli, l'ID del AMI proprietario, lo stato della configurazione di avvio rapido, il motivo per cui lo stato è stato modificato e l'ora in cui è avvenuta la modifica dello stato.

```
aws ec2 describe-fast-launch-images
```

**Output:**



```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni sulla configurazione di Windows AMI per un avvio più rapido, consulta [Configure your AMI for faster launching](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta Command [DescribeFastLaunchImages](#)Reference AWS CLI .

## describe-fast-snapshot-restores

Il seguente esempio di codice mostra come utilizzare `describe-fast-snapshot-restores`.

### AWS CLI

Per descrivere i ripristini rapidi delle istantanee

L'esempio seguente visualizza i dettagli di tutti i ripristini rapidi di istantanee con uno stato di `disabled`

```
aws ec2 describe-fast-snapshot-restores \
  --filters Name=state,Values=disabled
```

Output:

```
{
  "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"
    }
  ]
}
```

L'`describe-fast-snapshot-restores` esempio seguente descrive tutti i ripristini rapidi di istantanee.

```
aws ec2 describe-fast-snapshot-restores
```

- Per API i dettagli, vedere [DescribeFastSnapshotRestores](#) in AWS CLI Command Reference.

## describe-fleet-history

Il seguente esempio di codice mostra come utilizzare `describe-fleet-history`.

### AWS CLI

Per descrivere la storia EC2 della flotta

L'`describe-fleet-history` esempio seguente restituisce la cronologia della EC2 flotta specificata a partire dall'ora specificata. L'output è per un EC2 Fleet con due istanze in esecuzione.

```
aws ec2 describe-fleet-history \
  --fleet-id fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
  --start-time 2020-09-01T00:00:00Z
```

## Output:

```
{
  "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",
        "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"  
}
```

Per ulteriori informazioni, consulta [Managing an EC2 Fleet](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

- Per API i dettagli, consulta AWS CLI Command [DescribeFleetHistory](#) Reference.

## describe-fleet-instances

Il seguente esempio di codice mostra come utilizzare `describe-fleet-instances`.

### AWS CLI

Per descrivere le istanze in esecuzione per una flotta EC2

L'esempio seguente descrive le istanze in esecuzione per il Fleet specificato EC2.

```
aws ec2 describe-fleet-instances \  
  --fleet-id 12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Managing an EC2 Fleet](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

- Per API i dettagli, consulta AWS CLI Command [DescribeFleetInstances](#) Reference.

## describe-fleets

Il seguente esempio di codice mostra come utilizzare `describe-fleets`.

### AWS CLI

Per descrivere una EC2 flotta

L'`describe-fleets` esempio seguente descrive la EC2 flotta specificata.

```
aws ec2 describe-fleets \  
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE
```

Output:

```
{  
  "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"
    }
  }
]
}
```

Per ulteriori informazioni, consulta [Managing an EC2 Fleet](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

- Per API i dettagli, consulta AWS CLI Command [DescribeFleets](#) Reference.

## describe-flow-logs

Il seguente esempio di codice mostra come utilizzare `describe-flow-logs`.

AWS CLI

Esempio 1: per descrivere tutti i log di flusso

L'`describe-flow-logs` esempio seguente mostra i dettagli di tutti i log di flusso.

```
aws ec2 describe-flow-logs
```

Output:

```
{
  "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}"
    }
  ]
}

```

Esempio 2: Per descrivere un sottoinsieme dei log di flusso

L' `describe-flow-logs` seguente utilizza un filtro per visualizzare i dettagli solo per i log di flusso che si trovano nel gruppo di log specificato in Amazon CloudWatch Logs.

```

aws ec2 describe-flow-logs \
  --filter "Name=log-group-name,Values=MyFlowLogs"

```

- Per API i dettagli, consulta AWS CLI Command [DescribeFlowLogsReference](#).

## describe-fpga-image-attribute

Il seguente esempio di codice mostra come utilizzare `describe-fpga-image-attribute`.

### AWS CLI

Per descrivere gli attributi di un'FPGA immagine Amazon

Questo esempio descrive le autorizzazioni di caricamento per l'oggetto specificato AFI.

Comando:

```
aws ec2 describe-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --  
attribute LoadPermission
```

Output:

```
{  
  "FpgaImageAttribute": {  
    "FpgaImageId": "afi-0d123e123bfc85abc",  
    "LoadPermissions": [  
      {  
        "UserId": "123456789012"  
      }  
    ]  
  }  
}
```

- Per API i dettagli, vedere [DescribeFpgaImageAttribute](#) in AWS CLI Command Reference.

## describe-fpga-images

Il seguente esempio di codice mostra come utilizzare `describe-fpga-images`.

### AWS CLI

Per descrivere le FPGA immagini di Amazon

Questo esempio descrive AFIs che sono di proprietà per account123456789012.

Comando:



```
aws ec2 describe-fpga-images --filters Name=owner-id,Values=123456789012
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeFpgaImages](#) in AWS CLI Command Reference.

## describe-host-reservation-offerings

Il seguente esempio di codice mostra come utilizzare `describe-host-reservation-offerings`.

### AWS CLI

Per descrivere le offerte di prenotazione dedicata agli host

Questo esempio descrive le prenotazioni di host dedicati per la famiglia di istanze M4 disponibili per l'acquisto.

## Comando:

```
aws ec2 describe-host-reservation-offerings --filter Name=instance-family,Values=m4
```

## Output:

```
{
  "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
  }
]
```

- Per API i dettagli, vedere [DescribeHostReservationOfferings](#) in AWS CLI Command Reference.

## describe-host-reservations

Il seguente esempio di codice mostra come utilizzare `describe-host-reservations`.

### AWS CLI

Per descrivere Dedicated Host Reservations nel tuo account

Questo esempio descrive le prenotazioni dedicate agli host presenti nel tuo account.

Comando:

```
aws ec2 describe-host-reservations
```

Output:

```
{
  "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
  }
]
}

```

- Per API i dettagli, consulta [DescribeHostReservations AWS CLI Command Reference](#).

## describe-hosts

Il seguente esempio di codice mostra come utilizzare `describe-hosts`.

### AWS CLI

Per visualizzare i dettagli sugli host dedicati

L'esempio seguente mostra i dettagli degli host `available` dedicati presenti nel tuo AWS account.

```
aws ec2 describe-hosts --filter "Name=state,Values=available"
```

Output:

```

{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione degli host dedicati](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per API i dettagli, consulta AWS CLI Command [DescribeHosts](#) Reference.

## **describe-iam-instance-profile-associations**

Il seguente esempio di codice mostra come utilizzare `describe-iam-instance-profile-associations`.

### AWS CLI

Per descrivere le associazioni dei profili di IAM istanza

Questo esempio descrive tutte le associazioni dei profili di IAM istanza.

Comando:

```
aws ec2 describe-iam-instance-profile-associations
```

Output:

```
{
  "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"
      }
    }
  ]
}
```

- Per API i dettagli, consulta [DescribeIamInstanceProfileAssociations AWS CLI Command Reference](#).

## describe-id-format

Il seguente esempio di codice mostra come utilizzare `describe-id-format`.

AWS CLI

Esempio 1: Per descrivere il formato ID di una risorsa

L'`describe-id-format`esempio seguente descrive il formato ID per i gruppi di sicurezza.

```
aws ec2 describe-id-format \
```

```
--resource security-group
```

Nell'output di esempio seguente, il Deadline valore indica che la scadenza per il passaggio definitivo di questo tipo di risorsa dal formato ID breve a quello ID lungo è scaduta alle 00:00 del 15 UTC agosto 2018.

```
{
  "Statuses": [
    {
      "Deadline": "2018-08-15T00:00:00.000Z",
      "Resource": "security-group",
      "UseLongIds": true
    }
  ]
}
```

Esempio 2: per descrivere il formato ID per tutte le risorse

L'`describe-id-format` esempio seguente descrive il formato ID per tutti i tipi di risorse. Tutti i tipi di risorse che supportavano il formato ID breve sono stati convertiti al formato ID lungo.

```
aws ec2 describe-id-format
```

- Per API i dettagli, vedere [DescribeIdFormat](#) in AWS CLI Command Reference.

## describe-identity-id-format

Il seguente esempio di codice mostra come utilizzare `describe-identity-id-format`.

### AWS CLI

Per descrivere il formato ID di un IAM ruolo

L'`describe-identity-id-format` esempio seguente descrive il formato ID ricevuto dalle istanze create dal IAM ruolo `EC2Role` nell' AWS account.

```
aws ec2 describe-identity-id-format \
  --principal-arn arn:aws:iam::123456789012:role/my-iam-role \
  --resource instance
```

L'output seguente indica che le istanze create da questo ruolo ricevono IDs in formato ID lungo.

```
{
  "Statuses": [
    {
      "Deadline": "2016-12-15T00:00:00Z",
      "Resource": "instance",
      "UseLongIds": true
    }
  ]
}
```

Per descrivere il formato ID di un IAM utente

L'`describe-identity-id-format` seguente descrive il formato ID ricevuto dalle istantanee create dall'IAM utente `AdminUser` nell'AWS account.

```
aws ec2 describe-identity-id-format \
  --principal-arn arn:aws:iam::123456789012:user/AdminUser \
  --resource snapshot
```

L'output indica che le istantanee create da questo utente vengono ricevute IDs in formato ID lungo.

```
{
  "Statuses": [
    {
      "Deadline": "2016-12-15T00:00:00Z",
      "Resource": "snapshot",
      "UseLongIds": true
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeIdentityIdFormat](#) in AWS CLI Command Reference.

## **describe-image-attribute**

Il seguente esempio di codice mostra come utilizzare `describe-image-attribute`.

### AWS CLI

Per descrivere le autorizzazioni di avvio per un AMI



Questo esempio descrive le autorizzazioni di avvio per il file specificato. AMI

Comando:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --  
attribute LaunchPermission
```

Output:

```
{  
  "LaunchPermissions": [  
    {  
      "UserId": "123456789012"  
    }  
  ],  
  "ImageId": "ami-5731123e",  
}
```

Per descrivere i codici di prodotto per un AMI

Questo esempio descrive i codici prodotto per quanto specificato AMI. Tieni presente che questo non AMI ha codici di prodotto.

Comando:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --attribute productCodes
```

Output:

```
{  
  "ProductCodes": [],  
  "ImageId": "ami-5731123e",  
}
```

- Per API i dettagli, vedere [DescribeImageAttribute](#) in AWS CLI Command Reference.

## describe-images

Il seguente esempio di codice mostra come utilizzare `describe-images`.

## AWS CLI

### Esempio 1: Per descrivere un AMI

L'output di `aws ec2 describe-images` seguente descrive quanto specificato AMI nella regione specificata.

```
aws ec2 describe-images \
  --region us-east-1 \
  --image-ids ami-1234567890EXAMPLE
```

### Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Amazon Machine Images \(AMI\)](#) nella Amazon EC2 User Guide.

Esempio 2: descrivere AMIs in base ai filtri

L'`describe-images`esempio seguente descrive Windows AMIs fornito da Amazon e supportato da AmazonEBS.

```

aws ec2 describe-images \
  --owners amazon \
  --filters "Name=platform,Values=windows" "Name=root-device-type,Values=ebs"

```

Per un esempio dell'output di `describe-images`, vedi l'Esempio 1.

Per ulteriori esempi di utilizzo dei filtri, consulta [Elencare e filtrare le risorse](#) nella Amazon EC2 User Guide.

Esempio 3: descrivere in AMIs base ai tag

L'`describe-images`esempio seguente descrive tutto ciò AMIs che ha il `tagType=Custom`. L'esempio utilizza il `--query` parametro per visualizzare solo il AMIIDs.

```

aws ec2 describe-images \
  --filters "Name=tag:Type,Values=Custom" \
  --query 'Images[*].[ImageId]' \
  --output text

```

Output:

```

ami-1234567890EXAMPLE
ami-0abcdef1234567890

```

Per ulteriori esempi di utilizzo dei filtri di tag, consulta [Lavorare con i tag](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DescribeImages AWS CLI Command Reference](#).

## describe-import-image-tasks

Il seguente esempio di codice mostra come utilizzare `describe-import-image-tasks`.

### AWS CLI

Per monitorare un'operazione di importazione di immagini

L'esempio seguente controlla lo stato dell'attività di importazione dell'immagine specificata.

```
aws ec2 describe-import-image-tasks \
  --import-task-ids import-ami-1234567890abcdef0
```

Output per un'operazione di importazione di immagini in corso.

```
{
  "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"
    }
  ]
}
```

Output per un'operazione di importazione di immagini completata. L'ID del risultato AMI è fornito da `ImageId`.

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeImportImageTasks](#) in AWS CLI Command Reference.

## describe-import-snapshot-tasks

Il seguente esempio di codice mostra come utilizzare `describe-import-snapshot-tasks`.

### AWS CLI

Per monitorare un'operazione di importazione di istantanee

L'`describe-import-snapshot-tasks` esempio seguente controlla lo stato dell'attività di importazione dello snapshot specificata.

```
aws ec2 describe-import-snapshot-tasks \
  --import-task-ids import-snap-1234567890abcdef0
```

Output per un'operazione di importazione di istantanee in corso:

```
{
  "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/converting",
        "UserBucket": {
          "S3Bucket": "my-import-bucket",
          "S3Key": "vms/my-server-vm.vmdk"
        }
      }
    }
  ]
}

```

Output per un'attività di importazione di istantanee completata. L'ID dell'istananea risultante è fornito da. SnapshotId

```

{
  "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"
        }
      }
    }
  ]
}

```

- Per API i dettagli, vedere [DescribeImportSnapshotTasks](#) in AWS CLI Command Reference.

## describe-instance-attribute

Il seguente esempio di codice mostra come utilizzare `describe-instance-attribute`.

### AWS CLI

Per descrivere il tipo di istanza

Questo esempio descrive il tipo di istanza dell'istanza specificata.

Comando:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute instanceType
```

Output:

```
{  
  "InstanceId": "i-1234567890abcdef0"  
  "InstanceType": {  
    "Value": "t1.micro"  
  }  
}
```

Per descrivere l' `disableApiTermination` attributo

Questo esempio descrive l'`disableApiTermination` attributo dell'istanza specificata.

Comando:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute disableApiTermination
```

Output:

```
{  
  "InstanceId": "i-1234567890abcdef0"  
  "DisableApiTermination": {  
    "Value": "false"  
  }  
}
```

Per descrivere la mappatura dei dispositivi a blocchi per un'istanza

Questo esempio descrive l'`blockDeviceMapping` attributo dell'istanza specificata.

Comando:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute blockDeviceMapping
```

Output:

```
{  
  "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"  
      }  
    }  
  ],  
}
```

- Per API i dettagli, vedere [DescribeInstanceAttribute](#) in AWS CLI Command Reference.

## **describe-instance-connect-endpoints**

Il seguente esempio di codice mostra come utilizzare `describe-instance-connect-endpoints`.

AWS CLI

Per descrivere un endpoint EC2 Instance Connect



L'esempio seguente descrive l'endpoint EC2 Instance Connect specificato.

```
aws ec2 describe-instance-connect-endpoints \
  --region us-east-1 \
  --instance-connect-endpoint-ids eice-0123456789example
```

Output:

```
{
  "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": []
    }
  ]
}
```

Per ulteriori informazioni, consulta [Create an EC2 Instance Connect Endpoint](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DescribeInstanceConnectEndpoints AWS CLI Command Reference](#).

## describe-instance-credit-specifications

Il seguente esempio di codice mostra come utilizzare `describe-instance-credit-specifications`.

## AWS CLI

Descrivere l'opzione di credito per CPU l'utilizzo di una o più istanze

L'`describe-instance-credit-specifications` esempio seguente descrive l'opzione CPU di credito per l'istanza specificata.

```
aws ec2 describe-instance-credit-specifications \
  --instance-ids i-1234567890abcdef0
```

Output:

```
{
  "InstanceCreditSpecifications": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "CpuCredits": "unlimited"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Lavora con istanze a prestazioni espandibili](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeInstanceCreditSpecificationsReference](#).

## **describe-instance-event-notification-attributes**

Il seguente esempio di codice mostra come utilizzare `describe-instance-event-notification-attributes`.

## AWS CLI

Per descrivere i tag per le notifiche degli eventi programmati

L'`describe-instance-event-notification-attributes` esempio seguente descrive i tag da visualizzare nelle notifiche degli eventi pianificati.

```
aws ec2 describe-instance-event-notification-attributes
```

Output:

```
{
  "InstanceTagAttribute": {
    "InstanceTagKeys": [],
    "IncludeAllTagsOfInstance": true
  }
}
```

Per ulteriori informazioni, consulta [Eventi pianificati per le tue istanze](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per API i dettagli, consulta Command [DescribeInstanceEventNotificationAttributes](#)Reference AWS CLI .

## describe-instance-event-windows

Il seguente esempio di codice mostra come utilizzare `describe-instance-event-windows`.

### AWS CLI

Esempio 1: Per descrivere tutte le finestre degli eventi

L'`describe-instance-event-windows`esempio seguente descrive tutte le finestre degli eventi nella regione specificata.

```
aws ec2 describe-instance-event-windows \
  --region us-east-1
```

Output:

```
{
  "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"
}

```

Esempio 2: Per descrivere una finestra di eventi specifica

L'`describe-instance-event-windows` seguente descrive un evento specifico utilizzando il `instance-event-window` parametro per descrivere una finestra di eventi specifica.

```

aws ec2 describe-instance-event-windows \
  --region us-east-1 \
  --instance-event-window-ids iew-0abcdef1234567890

```

Output:

```

{
  "InstanceEventWindows": [
    {
      "InstanceEventWindowId": "iew-0abcdef1234567890",
      "Name": "myEventWindowName",
      "CronExpression": "* 21-23 * * 2,3",
      "AssociationTarget": {
        "InstanceIds": [
          "i-1234567890abcdef0",
          "i-0598c7d356eba48d7"
        ],
        "Tags": [],
        "DedicatedHostIds": []
      },
      "State": "active",
      "Tags": []
    }
  ]
}

```

### Esempio 3: Per descrivere le finestre degli eventi che corrispondono a uno o più filtri

L'`describe-instance-event-windows` seguente descrive le finestre degli eventi che corrispondono a uno o più filtri utilizzando il `filter` parametro. Il `instance-id` filtro viene utilizzato per descrivere tutte le finestre degli eventi associate all'istanza specificata. Quando viene utilizzato un filtro, si stabilisce una corrispondenza diretta. Tuttavia, il filtro `instance-id` è diverso. Se non esiste una corrispondenza diretta con l'ID dell'istanza, si ricorre alle associazioni indirette con la finestra dell'evento, ad esempio i tag dell'istanza o l'ID host dedicato (se l'istanza è un host dedicato).

```
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>
```

#### Output:

```
{
  "InstanceEventWindows": [
    {
      "InstanceEventWindowId": "iew-0dbc0adb66f235982",
      "TimeRanges": [
        {
          "StartWeekDay": "sunday",
          "StartHour": 2,
          "EndWeekDay": "sunday",
          "EndHour": 8
        }
      ],
      "Name": "myEventWindowName",
      "AssociationTarget": {
        "InstanceIds": [],
        "Tags": [],
        "DedicatedHostIds": [
          "h-0140d9a7ecbd102dd"
        ]
      },
      "State": "active",
      "Tags": []
    }
  ]
}
```

```
}
```

Nell'output di esempio, l'istanza si trova su un host dedicato, associato alla finestra dell'evento.

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta Command [DescribeInstanceEventWindows](#)Reference AWS CLI .

## describe-instance-status

Il seguente esempio di codice mostra come utilizzare `describe-instance-status`.

### AWS CLI

Per descrivere lo stato delle istanze

Nell'esempio di `describe-instance-status` seguente viene descritto lo stato attuale dell'istanza specificata.

```
aws ec2 describe-instance-status \  
  --instance-ids i-1234567890abcdef0
```

Output:

```
{  
  "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"
        }
      ]
    }
  ]
}

```

Per ulteriori informazioni, consulta [Monitora lo stato delle tue istanze](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DescribeInstanceStatus AWS CLI Command Reference](#).

## describe-instance-topology

Il seguente esempio di codice mostra come utilizzare `describe-instance-topology`.

### AWS CLI

Per descrivere la topologia delle istanze di tutte le istanze

L'esempio seguente descrive la topologia di tutte le istanze che corrispondono ai tipi di istanze supportati per questo comando.

```

aws ec2 describe-instance-topology \
  --region us-west-2

```

Output:

```

{
  "Instances": [
    {
      "InstanceId": "i-111111111example",
      "InstanceType": "p4d.24xlarge",
      "GroupName": "my-ml-cpg",
      "NetworkNodes": [
        "nn-111111111example",
        "nn-222222222example",

```

```
        "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-4444444444example",
    "InstanceType": "trn1.2xlarge",
    "NetworkNodes": [
        "nn-1111111111example",
        "nn-5434334334example",
        "nn-1235301234example"
    ],
    "ZoneId": "usw2-az2",
    "AvailabilityZone": "us-west-2a"
}
],
"NextToken": "SomeEncryptedToken"
}
```



Per ulteriori informazioni, inclusi altri esempi, consulta la [topologia delle EC2 istanze Amazon](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DescribeInstanceTopology AWS CLI Command Reference](#).

## describe-instance-type-offerings

Il seguente esempio di codice mostra come utilizzare `describe-instance-type-offerings`.

### AWS CLI

Esempio 1: per elencare i tipi di istanze offerti in una regione

L'`describe-instance-type-offerings` esempio seguente elenca i tipi di istanze offerti nella regione configurata come regione predefinita per AWS CLI.

```
aws ec2 describe-instance-type-offerings
```

Per elencare i tipi di istanze offerti in una regione diversa, specifica la regione utilizzando il `--region` parametro.

```
aws ec2 describe-instance-type-offerings \  
  --region us-east-2
```

Output:

```
{  
  "InstanceTypeOfferings": [  
    {  
      "InstanceType": "m5.2xlarge",  
      "LocationType": "region",  
      "Location": "us-east-2"  
    },  
    {  
      "InstanceType": "t3.micro",  
      "LocationType": "region",  
      "Location": "us-east-2"  
    },  
    ...  
  ]  
}
```

Esempio 2: per elencare i tipi di istanze offerti in una zona di disponibilità

L'`describe-instance-type-offerings` comando seguente elenca i tipi di istanze offerti nella zona di disponibilità specificata. La zona di disponibilità deve trovarsi nella regione specificata.

```
aws ec2 describe-instance-type-offerings \  
  --location-type availability-zone \  
  --filters Name=location,Values=us-east-2a \  
  --region us-east-2
```

Esempio 3: per verificare se un tipo di istanza è supportato

Il `describe-instance-type-offerings` comando seguente indica se il tipo di `c5.xlarge` istanza è supportato nella regione specificata.

```
aws ec2 describe-instance-type-offerings \  
  --filters Name=instance-type,Values=c5.xlarge \  
  --region us-east-2
```

L'`describe-instance-type-offerings` comando seguente elenca tutti i tipi di istanza C5 supportati nella regione specificata.

```
aws ec2 describe-instance-type-offerings \  
  --filters Name=instance-type,Values=c5* \  
  --query "InstanceTypeOfferings[].InstanceType" \  
  --region us-east-2
```

Output:

```
[  
  "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"
]
```

- Per API i dettagli, vedere [DescribeInstanceTypeOfferings](#) in AWS CLI Command Reference.

## describe-instance-types

Il seguente esempio di codice mostra come utilizzare `describe-instance-types`.

### AWS CLI

Esempio 1: per descrivere un tipo di istanza

Nell'esempio di `describe-instance-types` seguente vengono visualizzati i dettagli del tipo di istanza specificato.

```
aws ec2 describe-instance-types \
  --instance-types t2.micro
```

Output:

```
{
  "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
  }
]
}
```

Per ulteriori informazioni, consulta la Guida per l'utente dei [tipi di istanze](#) di Amazon Elastic Compute Cloud per le istanze Linux.

Esempio 2: per filtrare i tipi di istanza disponibili

È possibile specificare un filtro per rifinire i risultati in base ai tipi di istanza che hanno una caratteristica specifica. Nell'esempio di `describe-instance-types` seguente vengono elencati i tipi di istanza che supportano l'ibernazione.

```
aws ec2 describe-instance-types \
  --filters Name=hibernation-supported,Values=true --query
  'InstanceTypes[*].InstanceType'
```

Output:

```
[
  "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"
]
```

Per ulteriori informazioni, consulta la Guida per l'utente dei [tipi di istanze](#) di Amazon Elastic Compute Cloud per le istanze Linux.

- Per API i dettagli, consulta AWS CLI Command [DescribeInstanceTypes](#) Reference.

## describe-instances

Il seguente esempio di codice mostra come utilizzare `describe-instances`.

### AWS CLI

Esempio 1: per descrivere un'istanza

Nell'esempio di `describe-instances` seguente viene descritta l'istanza specificata.

```
aws ec2 describe-instances \  
  --instance-ids i-1234567890abcdef0
```

Output:

```
{  
  "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": "11111111111111111111"
    },
    "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"
    }
]
}
```

Esempio 2: per filtrare per istanze secondo il tipo specificato

Nell'esempio di `describe-instances` seguente vengono utilizzati filtri per rifinire i risultati in base alle istanze del tipo specificato.

```
aws ec2 describe-instances \
  --filters Name=instance-type,Values=m5.Large
```

Per un output di esempio, vedi l'Esempio 1.

Per ulteriori informazioni, consulta [Elenca e filtra utilizzando la CLI](#) Guida per gli EC2 utenti di Amazon.

Esempio 3: per filtrare per istanze secondo il tipo e la zona di disponibilità specificati

Nell'esempio di `describe-instances` seguente vengono utilizzati più filtri per rifinire i risultati in base alle istanze del tipo specificato che si trovano anche nella zona di disponibilità specificata.

```
aws ec2 describe-instances \
  --filters Name=instance-type,Values=t2.micro,t3.micro Name=availability-zone,Values=us-east-2c
```

Per un output di esempio, vedi l'Esempio 1.

Esempio 4: per filtrare le istanze con il tipo e la zona di disponibilità specificati utilizzando un file JSON

L'`describe-instances` seguente utilizza un file JSON di input per eseguire lo stesso filtro dell'esempio precedente. Quando i filtri diventano più complicati, può essere più facile specificarli in un JSON file.

```
aws ec2 describe-instances \
  --filters file://filters.json
```

## Contenuto di `filters.json`.

```
[
  {
    "Name": "instance-type",
    "Values": ["t2.micro", "t3.micro"]
  },
  {
    "Name": "availability-zone",
    "Values": ["us-east-2c"]
  }
]
```

Per un output di esempio, vedi l'Esempio 1.

### Esempio 5: per filtrare per istanze secondo il tag Proprietario specificato

Nell'esempio di `describe-instances` seguente vengono utilizzati filtri tag per rifinire i risultati in base alle istanze che hanno un tag con la chiave tag specificata (Proprietario), a prescindere dal valore del tag.

```
aws ec2 describe-instances \
  --filters "Name=tag-key,Values=Owner"
```

Per un output di esempio, vedi l'Esempio 1.

### Esempio 6: per filtrare per istanze secondo il valore tag my-team specificato

Nell'esempio di `describe-instances` seguente vengono utilizzati filtri tag per rifinire i risultati in base alle istanze che hanno un tag con il valore tag specificato (my-team), a prescindere dalla chiave del tag.

```
aws ec2 describe-instances \
  --filters "Name=tag-value,Values=my-team"
```

Per un output di esempio, vedi l'Esempio 1.

### Esempio 7: per filtrare le istanze secondo i valori tag Proprietario e my-team specificati

Nell'esempio di `describe-instances` seguente vengono utilizzati filtri tag per rifinire i risultati in base alle istanze che hanno il tag specificato (Proprietario=my-team).

```
aws ec2 describe-instances \  
  --filters "Name=tag:Owner,Values=my-team"
```

Per un output di esempio, vedi l'Esempio 1.

Esempio 8: Per visualizzare solo l'istanza e la sottorete IDs per tutte le istanze

`describe-instances` Gli esempi seguenti utilizzano il `--query` parametro per visualizzare solo l'istanza e la sottorete IDs per tutte le istanze, in formato. JSON

Linux e 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
```

Output:

```
[  
  {  
    "Instance": "i-057750d42936e468a",  
    "Subnet": "subnet-069beee9b12030077"  
  },  
  {  
    "Instance": "i-001efd250faaa6ffa",  
    "Subnet": "subnet-0b715c6b7db68927a"  
  },  
  {  
    "Instance": "i-027552a73f021f3bd",  
    "Subnet": "subnet-0250c25a1f4e15235"  
  }  
  ...  
]
```

Esempio 9: per filtrare le istanze del tipo specificato e visualizzare solo le relative istanze IDs

L'`describe-instances` seguente utilizza i filtri per limitare i risultati alle istanze del tipo specificato e il `--query` parametro per visualizzare solo l'istanza. IDs

```
aws ec2 describe-instances \
  --filters "Name=instance-type,Values=t2.micro" \
  --query "Reservations[*].Instances[*].[InstanceId]" \
  --output text
```

Output:

```
i-031c0dc19de2fb70c
i-00d8bfff789a736b75
i-0b715c6b7db68927a
i-0626d4edd54f1286d
i-00b8ae04f9f99908e
i-0fc71c25d2374130c
```

Esempio 10: per filtrare le istanze del tipo specificato e visualizzare solo la relativa istanzaIDs, la zona di disponibilità e il valore del tag specificato

Negli esempi di `describe-instances` seguenti vengono visualizzati l'ID dell'istanza, la zona di disponibilità e il valore del tag Name per le istanze che hanno un tag con il nome tag-key, in formato tabella.

Linux e 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
```

Output:

```

-----
|                               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  |
+-----+-----+-----+-----+

```

Esempio 11: per descrivere le istanze in un gruppo di posizionamento delle partizioni

Nell'esempio di `describe-instances` seguente viene descritta l'istanza specificata. L'output include le informazioni di collocamento dell'istanza, che a loro volta comprendono il nome del gruppo di collocamento e il numero di partizioni per l'istanza.

```

aws ec2 describe-instances \
  --instance-ids i-0123a456700123456 \
  --query "Reservations[*].Instances[*].Placement"

```

Output:

```

[
  [
    {
      "AvailabilityZone": "us-east-1c",
      "GroupName": "HDFS-Group-A",
      "PartitionNumber": 3,
      "Tenancy": "default"
    }
  ]
]

```

Per ulteriori informazioni, consulta [Descrivere le istanze in un gruppo di collocamento](#) nella Amazon EC2 User Guide.

Esempio 12: per filtrare le istanze secondo il gruppo di collocamento e il numero di partizioni specificati

Nell'esempio di `describe-instances` seguente i risultati vengono filtrati solamente in base alle istanze con il gruppo di collocamento e il numero di partizioni specificati.

```
aws ec2 describe-instances \  
  --filters "Name=placement-group-name,Values=HDFS-Group-A" "Name=placement-  
partition-number,Values=7"
```

Di seguito vengono mostrate solo le informazioni rilevanti contenute nell'output.

```
"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"  
    }  
  },  
]
```

Per ulteriori informazioni, consulta [Descrivere le istanze in un gruppo di collocamento](#) nella Amazon EC2 User Guide.

Esempio 13: per filtrare le istanze configurate per consentire l'accesso ai tag dai metadati dell'istanza

Nell'esempio di `describe-instances` seguente i risultati vengono filtrati solamente in base alle istanze configurate per consentire l'accesso ai tag dell'istanza dai metadati dell'istanza stessa.

```
aws ec2 describe-instances \  
  --filters "Name=tag:tag-key,Values=tag-value" "Name=tag:tag-key,Values=tag-value"
```

```
--filters "Name=metadata-options.instance-metadata-tags,Values=enabled" \  
--query "Reservations[*].Instances[*].InstanceId" \  
--output text
```

Di seguito è riportato l'output previsto.

```
i-1234567890abcdefg  
i-abcdefg1234567890  
i-1111111111aaaaaaaa  
i-aaaaaaaa1111111111
```

Per ulteriori informazioni, consulta [Lavora con i tag di istanza nei metadati delle istanze](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DescribeInstances AWS CLI](#) Command Reference.

## describe-internet-gateways

Il seguente esempio di codice mostra come utilizzare `describe-internet-gateways`.

### AWS CLI

Per descrivere un gateway Internet

L'`describe-internet-gateways` esempio seguente descrive il gateway Internet specificato.

```
aws ec2 describe-internet-gateways \  
--internet-gateway-ids igw-0d0fb496b3EXAMPLE
```

Output:

```
{  
  "InternetGateways": [  
    {  
      "Attachments": [  
        {  
          "State": "available",  
          "VpcId": "vpc-0a60eb65b4EXAMPLE"  
        }  
      ],  
      "InternetGatewayId": "igw-0d0fb496b3EXAMPLE",
```



```
    "OwnerId": "123456789012",
    "Tags": [
      {
        "Key": "Name",
        "Value": "my-igw"
      }
    ]
  }
]
```

Per ulteriori informazioni, consulta [Internet gateway](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [DescribeInternetGateways AWS CLI Command Reference](#).

## describe-ipam-pools

Il seguente esempio di codice mostra come utilizzare `describe-ipam-pools`.

### AWS CLI

Per visualizzare i dettagli di un IPAM pool

L'esempio seguente mostra i dettagli dei pool.

(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
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [DescrivelipamPools](#) in AWS CLI Command Reference.

## **describe-ipam-resource-discoveries**

Il seguente esempio di codice mostra come utilizzare `describe-ipam-resource-discoveries`.

### AWS CLI

Esempio 1: Visualizza i dettagli completi delle scoperte di risorse

In questo esempio, sei un IPAM amministratore delegato che desidera creare e condividere una ricerca di risorse con l'IPAM amministratore di un'altra AWS organizzazione in modo che l'amministratore possa gestire e monitorare gli indirizzi IP delle risorse della tua organizzazione.

Questo esempio può essere utile se:

Hai provato a creare una scoperta di risorse, ma hai ricevuto un errore indicante che hai raggiunto il limite di 1. Ti rendi conto che potresti aver già creato una ricerca di risorse e desideri visualizzarla nel tuo account. In una regione sono presenti risorse che non vengono scoperte da IPAM. Desideri visualizzare le informazioni `--operating-regions` definite per la risorsa e assicurarti di aver aggiunto la regione corretta come regione operativa in modo che le risorse presenti possano essere scoperte.

L'`describe-ipam-resource-discoveries` esempio seguente elenca i dettagli della scoperta delle risorse nel tuo AWS account. È possibile effettuare una sola ricerca di risorse per AWS regione.

```
aws ec2 describe-ipam-resource-discoveries \
  --region us-east-1
```

Output:

```
{
  "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": []
    }
  ]
}
```

Per ulteriori informazioni, consulta [Integrazione IPAM con account esterni alla tua organizzazione](#) nella Amazon VPC IPAM User Guide.

Esempio 2: Visualizza solo l'individuazione delle risorse IDs

L'`describe-ipam-resource-discoveries` seguente elenca l'ID della scoperta delle risorse nel tuo AWS account. È possibile effettuare una sola ricerca di risorse per AWS regione.

```
aws ec2 describe-ipam-resource-discoveries \
  --query "IpamResourceDiscoveries[*].IpamResourceDiscoveryId" \
  --output text
```

Output:

```
ipam-res-disco-0481e39b242860333
```

Per ulteriori informazioni, consulta [Integrazione IPAM con account esterni alla tua organizzazione](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [DescribeIpamResourceDiscoveries AWS CLI Command Reference](#).

## **describe-ipam-resource-discovery-associations**

Il seguente esempio di codice mostra come utilizzare `describe-ipam-resource-discovery-associations`.

### AWS CLI

Per visualizzare tutte le associazioni di ricerca delle risorse con il IPAM

In questo esempio, sei un amministratore IPAM delegato che ha associato Resource Discovery al tuo account IPAM per integrare altri account con il tuo. IPAM Hai notato che non stai scoprendo le risorse nelle aree operative di scoperta delle risorse come previsto. IPAM Vuoi controllare lo stato e lo stato della scoperta delle risorse per assicurarti che l'account che l'ha creata sia ancora attivo e che la scoperta delle risorse sia ancora condivisa.

`--region` Deve essere la tua regione di origine IPAM.

L'`describe-ipam-resource-discovery-associations` seguente elenca le associazioni di ricerca delle risorse nel tuo AWS account.

```
aws ec2 describe-ipam-resource-discovery-associations \
  --region us-east-1
```

Output:

```
{
  "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 questo esempio, dopo aver eseguito questo comando, si nota che è disponibile una funzione di rilevamento delle risorse non predefinita ("IsDefault": false ``) that is

```"ResourceDiscoveryStatus": "not-found"and"State": "create-complete".`  
 L'account del proprietario di Resource Discovery è stato chiuso. Se, in un altro caso, si nota che è `"ResourceDiscoveryStatus": "not-found" e"State": "associate-complete"`, ciò indica che si è verificata una delle seguenti situazioni:

L'individuazione delle risorse è stata eliminata dal proprietario dell'individuazione delle risorse. Il proprietario dell'individuazione delle risorse ha annullato la condivisione della scoperta della risorsa.

Per ulteriori informazioni, consulta [Integrazione IPAM con account esterni alla tua organizzazione](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [DescribeIpamResourceDiscoveryAssociations AWS CLI Command Reference](#).

## describe-ipam-scopes

Il seguente esempio di codice mostra come utilizzare `describe-ipam-scopes`.

### AWS CLI

Per visualizzare i dettagli di un IPAM cannocchiale

L'`describe-ipam-scopes` esempio seguente mostra i dettagli degli ambiti.

```
aws ec2 describe-ipam-scopes \
  --filters Name=owner-id,Values=123456789012 Name=ipam-
  id,Values=ipam-08440e7a3acde3908
```

Output:

```
{
  "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"
      }
    ]
  }
]
}

```

- Per API i dettagli, vedere [DescrìbelpamScopes](#) in AWS CLI Command Reference.

## describe-ipams

Il seguente esempio di codice mostra come utilizzare `describe-ipams`.

### AWS CLI

Per visualizzare i dettagli di un IPAM

L'esempio seguente mostra i dettagli di un IPAM.

```
aws ec2 describe-ipams \
  --filters Name=owner-id,Values=123456789012
```

Output:

```
{
  "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"
        }
      ]
    }
  ]
}
```



```

    }
  ]
}

```

- Per API i dettagli, vedere [DescribePools](#) in AWS CLI Command Reference.

## describe-ipv6-pools

Il seguente esempio di codice mostra come utilizzare `describe-ipv6-pools`.

### AWS CLI

Per descrivere i tuoi pool di IPv6 indirizzi

L'`describe-ipv6-pools` esempio seguente mostra i dettagli di tutti i tuoi pool di IPv6 indirizzi.

```
aws ec2 describe-ipv6-pools
```

Output:

```

{
  "Ipv6Pools": [
    {
      "PoolId": "ipv6pool-ec2-012345abc12345abc",
      "PoolCidrBlocks": [
        {
          "Cidr": "2001:db8:123::/48"
        }
      ],
      "Tags": [
        {
          "Key": "pool-1",
          "Value": "public"
        }
      ]
    }
  ]
}

```

- Per API i dettagli, vedere [DescribeIpv6Pools](#) in AWS CLI Command Reference.

## describe-key-pairs

Il seguente esempio di codice mostra come utilizzare `describe-key-pairs`.

### AWS CLI

Per visualizzare una coppia di chiavi

Nell'esempio di `describe-key-pairs` seguente vengono visualizzate informazioni sulla coppia di chiavi specificata.

```
aws ec2 describe-key-pairs \  
  --key-names my-key-pair
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Descrivi le chiavi pubbliche](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DescribeKeyPairs AWS CLI Command Reference](#).

## describe-launch-template-versions

Il seguente esempio di codice mostra come utilizzare `describe-launch-template-versions`.

### AWS CLI

Per descrivere le versioni dei modelli di lancio

Questo esempio descrive le versioni del modello di lancio specificato.

## Comando:

```
aws ec2 describe-launch-template-versions --launch-template-id lt-068f72b72934aff71
```

## Output:

```
{
  "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"
}
]
}

```

- Per API i dettagli, vedere [DescribeLaunchTemplateVersions](#) in AWS CLI Command Reference.

## describe-launch-templates

Il seguente esempio di codice mostra come utilizzare `describe-launch-templates`.

## AWS CLI

Per descrivere i modelli di lancio

Questo esempio descrive i tuoi modelli di lancio.

Comando:

```
aws ec2 describe-launch-templates
```

Output:

```
{
  "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"
  },
]
}

```

- Per API i dettagli, consulta [DescribeLaunchTemplates AWS CLI Command Reference](#).

## describe-local-gateway-route-table-virtual-interface-group-associations

Il seguente esempio di codice mostra come utilizzare `describe-local-gateway-route-table-virtual-interface-group-associations`.

### AWS CLI

Per descrivere le associazioni tra gruppi di interfacce virtuali e tabelle di routing del gateway locale

L'esempio seguente descrive le associazioni tra i gruppi di interfacce virtuali e le tabelle di routing del gateway locale presenti nell'AWS account.

```
aws ec2 describe-local-gateway-route-table-virtual-interface-group-associations
```

Output:

```

{
  "LocalGatewayRouteTableVirtualInterfaceGroupAssociations": [
    {
      "LocalGatewayRouteTableVirtualInterfaceGroupAssociationId": "lgw-vif-grp-assoc-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": []
    }
  ]
}

```

```
}

```

Per ulteriori informazioni, consulta [Lavorare con i gateway locali nella Guida per l'utente di AWS Outposts](#).

- Per API i dettagli, vedere [DescribeLocalGatewayRouteTableVirtualInterfaceGroupAssociations](#) in AWS CLI Command Reference.

## describe-local-gateway-route-table-vpc-associations

Il seguente esempio di codice mostra come utilizzare `describe-local-gateway-route-table-vpc-associations`.

### AWS CLI

Per descrivere le associazioni tra le tabelle di routing del gateway locale VPCs e le tabelle di routing

L'`describe-local-gateway-route-table-vpc-associations` esempio seguente visualizza informazioni sull'associazione specificata tra le tabelle di routing del gateway locale VPCs e quelle relative all'associazione specificata.

```
aws ec2 describe-local-gateway-route-table-vpc-associations \
  --local-gateway-route-table-vpc-association-ids lgw-vpc-assoc-0e0f27af15EXAMPLE
```

Output:

```
{
  "LocalGatewayRouteTableVpcAssociation": {
    "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0e0f27af1EXAMPLE",
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
    "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
    "VpcId": "vpc-0efe9bde08EXAMPLE",
    "State": "associated"
  }
}
```

Per ulteriori informazioni, consulta [Local gateway route tables](#) nella Guida per l'utente di Outposts.

- Per API i dettagli, vedere [DescribeLocalGatewayRouteTableVpcAssociations](#) in AWS CLI Command Reference.

## describe-local-gateway-route-tables

Il seguente esempio di codice mostra come utilizzare `describe-local-gateway-route-tables`.

### AWS CLI

Per descrivere le tabelle di routing del gateway locale

L'esempio seguente mostra i dettagli sulle tabelle di routing del gateway locale.

```
aws ec2 describe-local-gateway-route-tables
```

Output:

```
{
  "LocalGatewayRouteTables": [
    {
      "LocalGatewayRouteTableId": "lgw-rtb-059615ef7deEXAMPLE",
      "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
      "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/
op-0dc11b66edEXAMPLE",
      "State": "available"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeLocalGatewayRouteTables](#) in AWS CLI Command Reference.

## describe-local-gateway-virtual-interface-groups

Il seguente esempio di codice mostra come utilizzare `describe-local-gateway-virtual-interface-groups`.

### AWS CLI

Per descrivere i gruppi di interfacce virtuali del gateway locale

L'esempio seguente descrive i gruppi di interfaccia virtuale del gateway locale presenti nell'AWS account.



```
aws ec2 describe-local-gateway-virtual-interface-groups
```

Output:

```
{
  "LocalGatewayVirtualInterfaceGroups": [
    {
      "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
      "LocalGatewayVirtualInterfaceIds": [
        "lgw-vif-01a23bc4d5EXAMPLE",
        "lgw-vif-543ab21012EXAMPLE"
      ],
      "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
      "OwnerId": "123456789012",
      "Tags": []
    }
  ]
}
```

Per ulteriori informazioni, consulta [Lavorare con i gateway locali nella Guida per l'utente di AWS Outposts](#).

- Per API i dettagli, vedere [DescribeLocalGatewayVirtualInterfaceGroups](#) in AWS CLI Command Reference.

## describe-local-gateway-virtual-interfaces

Il seguente esempio di codice mostra come utilizzare `describe-local-gateway-virtual-interfaces`.

AWS CLI

Per descrivere le interfacce virtuali del gateway locale

L'esempio seguente descrive le interfacce virtuali del gateway locale del tuo AWS account.

```
aws ec2 describe-local-gateway-virtual-interfaces
```

Output:

```
{
  "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": []
    }
  ]
}
```

Per ulteriori informazioni, consulta [Lavorare con i gateway locali nella Guida per l'utente](#) di AWS Outposts.

- Per API i dettagli, vedere [DescribeLocalGatewayVirtualInterfaces](#) in AWS CLI Command Reference.

## describe-local-gateways

Il seguente esempio di codice mostra come utilizzare `describe-local-gateways`.

### AWS CLI

Per descrivere i gateway locali

L'esempio seguente mostra i dettagli dei gateway locali disponibili.

```
aws ec2 describe-local-gateways
```

Output:

```
{
  "LocalGateways": [
    {
      "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
      "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0dc11b66ed59f995a",
      "OwnerId": "123456789012",
      "State": "available"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeLocalGateways](#) in AWS CLI Command Reference.

## describe-locked-snapshots

Il seguente esempio di codice mostra come utilizzare `describe-locked-snapshots`.

AWS CLI

Per descrivere lo stato di blocco di un'istantanea

L'`describe-locked-snapshots` esempio seguente descrive lo stato di blocco dell'istantanea specificata.

```
aws ec2 describe-locked-snapshots \
  --snapshot-ids snap-0b5e733b4a8df6e0d
```

Output:

```
{
  "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"  
    }  
]  
}
```

Per ulteriori informazioni, consulta [Snapshot lock](#) nella Amazon EBS User Guide.

- Per API i dettagli, consulta [DescribeLockedSnapshots AWS CLI Command Reference](#).

## describe-managed-prefix-lists

Il seguente esempio di codice mostra come utilizzare `describe-managed-prefix-lists`.

### AWS CLI

Per descrivere gli elenchi di prefissi gestiti

L'esempio seguente descrive gli elenchi di prefissi di proprietà dell'account. AWS 123456789012

```
aws ec2 describe-managed-prefix-lists \  
--filters Name=owner-id,Values=123456789012
```

Output:

```
{  
  "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"
}
]
```

Per ulteriori informazioni, consulta [Managed Prefix Lists](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [DescribeManagedPrefixLists AWS CLI Command Reference](#).

## describe-moving-addresses

Il seguente esempio di codice mostra come utilizzare `describe-moving-addresses`.

### AWS CLI

Per descrivere i tuoi indirizzi di trasferimento

Questo esempio descrive tutti gli indirizzi IP elastici in movimento.

Comando:

```
aws ec2 describe-moving-addresses
```

Output:

```
{
  "MovingAddressStatuses": [
    {
      "PublicIp": "198.51.100.0",
      "MoveStatus": "MovingToVpc"
    }
  ]
}
```

```
}
```

Questo esempio descrive tutti gli indirizzi che vengono trasferiti sulla VPC piattaforma EC2 -.

Comando:

```
aws ec2 describe-moving-addresses --filters Name=moving-status,Values=MovingToVpc
```

- Per API i dettagli, vedere [DescribeMovingAddresses](#) in AWS CLI Command Reference.

## describe-nat-gateways

Il seguente esempio di codice mostra come utilizzare `describe-nat-gateways`.

AWS CLI

Esempio 1: Per descrivere un NAT gateway pubblico

L'`describe-nat-gateways` esempio seguente descrive il NAT gateway pubblico specificato.

```
aws ec2 describe-nat-gateways \  
  --nat-gateway-id nat-01234567890abcdef
```

Output:

```
{  
  "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"
  }
]
}

```

## Esempio 2: Per descrivere un NAT gateway privato

L'`describe-nat-gateways` esempio seguente descrive il NAT gateway privato specificato.

```

aws ec2 describe-nat-gateways \
  --nat-gateway-id nat-1234567890abcdef0

```

Output:

```

{
  "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"
}
]
}

```

Per ulteriori informazioni, consulta i [NATgateway](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [DescribeNatGateways AWS CLI Command Reference](#).

## describe-network-acls

Il seguente esempio di codice mostra come utilizzare `describe-network-acls`.

### AWS CLI

Per descrivere la rete ACLs

L'esempio seguente recupera i dettagli sulla rete ACLs.

```
aws ec2 describe-network-acls
```



## Output:

```
{
  "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"
}
]
}

```

Per ulteriori informazioni, vedere [Rete ACLs](#) nella Guida per l'AWS VPC utente.

- Per API i dettagli, vedere [DescribeNetworkAcls](#) in AWS CLI Command Reference.

## **describe-network-insights-access-scope-analyses**

Il seguente esempio di codice mostra come utilizzare `describe-network-insights-access-scope-analyses`.

### AWS CLI

Per descrivere le analisi dell'ambito di accesso a Network Insights

L'`describe-network-insights-access-scope-analyses` seguente descrive l'analisi dell'ambito di accesso nel tuo AWS account.

```
aws ec2 describe-network-insights-access-scope-analyses \
  --region us-east-1
```

Output:

```
{
  "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": []
    }
  ]
}
```

Per ulteriori informazioni, vedere [Guida introduttiva a Network Access Analyzer utilizzando la AWS CLI](#) guida Network Access Analyzer.

- Per API i dettagli, vedere [DescribeNetworkInsightsAccessScopeAnalyses](#) in AWS CLI Command Reference.

## `describe-network-insights-access-scopes`

Il seguente esempio di codice mostra come utilizzare `describe-network-insights-access-scopes`.

AWS CLI

Per descrivere gli ambiti di accesso a Network Insights

L'`describe-network-insights-access-scopes` seguente descrive le analisi dell'ambito di accesso nel tuo account. AWS

```
aws ec2 describe-network-insights-access-scopes \
```

```
--region us-east-1
```

Output:

```
{
  "NetworkInsightsAccessScopes": [
    {
      "NetworkInsightsAccessScopeId": "nis-123456789111",
      "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope/nis-123456789111",
      "CreateDate": "2021-11-29T21:12:41.416000+00:00",
      "UpdateDate": "2021-11-29T21:12:41.416000+00:00",
      "Tags": []
    }
  ]
}
```

Per ulteriori informazioni, consulta [Guida introduttiva a Network Access Analyzer utilizzando la AWS CLI guida](#) Network Access Analyzer.

- Per API i dettagli, vedere [DescribeNetworkInsightsAccessScopes](#) in AWS CLI Command Reference.

## describe-network-insights-analyses

Il seguente esempio di codice mostra come utilizzare `describe-network-insights-analyses`.

AWS CLI

Per visualizzare i risultati di un'analisi del percorso

L'`describe-network-insights-analyses` esempio seguente descrive l'analisi specificata. In questo esempio, l'origine è un gateway Internet, la destinazione è un'EC2istanza e il protocollo è TCP. L'analisi è riuscita (Status è `succeeded`) e il percorso non è raggiungibile (NetworkPathFound è `false`). Il codice esplicativo `ENI_SG_RULES_MISMATCH` indica che il gruppo di sicurezza per l'istanza non contiene una regola che consenta il traffico sulla porta di destinazione.

```
aws ec2 describe-network-insights-analyses \
  --network-insights-analysis-ids nia-02207aa13eb480c7a
```

## Output:

```
{
  "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": []
    }
  ]
}
```

```
}
```

Per ulteriori informazioni, consulta la [Guida introduttiva al AWS CLI Reachability Analyzer Guide](#).

- Per API i dettagli, vedere [DescribeNetworkInsightsAnalyses](#) in AWS CLI Command Reference.

## describe-network-insights-paths

Il seguente esempio di codice mostra come utilizzare `describe-network-insights-paths`.

### AWS CLI

Per descrivere un percorso

L'`describe-network-insights-paths` esempio seguente descrive il percorso specificato.

```
aws ec2 describe-network-insights-paths \  
  --network-insights-path-ids nip-0b26f224f1d131fa8
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta la [Guida introduttiva al AWS CLI Reachability Analyzer Guide](#).

- Per API i dettagli, vedere [DescribeNetworkInsightsPaths](#) in AWS CLI Command Reference.

## describe-network-interface-attribute

Il seguente esempio di codice mostra come utilizzare `describe-network-interface-attribute`.

### AWS CLI

Per descrivere l'attributo di allegato di un'interfaccia di rete

Questo comando di esempio descrive l'`attachment` attributo dell'interfaccia di rete specificata.

Comando:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --  
attribute attachment
```

Output:

```
{  
  "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"  
  }  
}
```

Per descrivere l'attributo `description` di un'interfaccia di rete

Questo comando di esempio descrive l'`description` attributo dell'interfaccia di rete specificata.

Comando:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --  
attribute description
```

Output:



```
{
  "NetworkInterfaceId": "eni-686ea200",
  "Description": {
    "Value": "My description"
  }
}
```

Per descrivere l' `groupSet` attributo di un'interfaccia di rete

Questo comando di esempio descrive l'`groupSet` attributo dell'interfaccia di rete specificata.

Comando:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute groupSet
```

Output:

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "Groups": [
    {
      "GroupName": "my-security-group",
      "GroupId": "sg-903004f8"
    }
  ]
}
```

Per descrivere l' `sourceDestCheck` attributo di un'interfaccia di rete

Questo comando di esempio descrive l'`sourceDestCheck` attributo dell'interfaccia di rete specificata.

Comando:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute sourceDestCheck
```

Output:

```
{
```

```
"NetworkInterfaceId": "eni-686ea200",
"SourceDestCheck": {
  "Value": true
}
}
```

- Per API i dettagli, vedere [DescribeNetworkInterfaceAttribute](#) in AWS CLI Command Reference.

## describe-network-interface-permissions

Il seguente esempio di codice mostra come utilizzare `describe-network-interface-permissions`.

### AWS CLI

Per descrivere le autorizzazioni dell'interfaccia di rete

Questo esempio descrive tutte le autorizzazioni dell'interfaccia di rete.

Comando:

```
aws ec2 describe-network-interface-permissions
```

Output:

```
{
  "NetworkInterfacePermissions": [
    {
      "PermissionState": {
        "State": "GRANTED"
      },
      "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",
      "NetworkInterfaceId": "eni-b909511a",
      "Permission": "INSTANCE-ATTACH",
      "AwsAccountId": "123456789012"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeNetworkInterfacePermissions](#) in AWS CLI Command Reference.

## describe-network-interfaces

Il seguente esempio di codice mostra come utilizzare `describe-network-interfaces`.

### AWS CLI

Per descrivere le interfacce di rete

Questo esempio descrive tutte le interfacce di rete.

Comando:

```
aws ec2 describe-network-interfaces
```

Output:

```
{
  "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"
  }
]
}
```

Questo esempio descrive le interfacce di rete che hanno un tag con la chiave `Purpose` e il valore `Prod`.

Comando:

```
aws ec2 describe-network-interfaces --filters Name=tag:Purpose,Values=Prod
```

Output:

```
{
  "NetworkInterfaces": [
    {
      "Status": "available",
      "MacAddress": "12:2c:bd:f9:bf:17",
      "SourceDestCheck": true,
```

```

    "VpcId": "vpc-8941ebec",
    "Description": "ProdENI",
    "NetworkInterfaceId": "eni-b9a5ac93",
    "PrivateIpAddresses": [
      {
        "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"
  }
]
}

```

- Per API i dettagli, vedere [DescribeNetworkInterfaces](#) in AWS CLI Command Reference.

## describe-placement-groups

Il seguente esempio di codice mostra come utilizzare `describe-placement-groups`.

## AWS CLI

Per descrivere i tuoi gruppi di collocamento

Questo comando di esempio descrive tutti i gruppi di collocamento.

Comando:

```
aws ec2 describe-placement-groups
```

Output:

```
{
  "PlacementGroups": [
    {
      "GroupName": "my-cluster",
      "State": "available",
      "Strategy": "cluster"
    },
    ...
  ]
}
```

- Per API i dettagli, consultate [DescribePlacementGroups AWS CLI Command Reference](#).

## describe-prefix-lists

Il seguente esempio di codice mostra come utilizzare `describe-prefix-lists`.

## AWS CLI

Per descrivere gli elenchi di prefissi

Questo esempio elenca tutti gli elenchi di prefissi disponibili per la regione.

Comando:

```
aws ec2 describe-prefix-lists
```

Output:

```
{
  "PrefixLists": [
```

```
{
  "PrefixListName": "com.amazonaws.us-east-1.s3",
  "Cidrs": [
    "54.231.0.0/17"
  ],
  "PrefixListId": "pl-63a5400a"
}
]
```

- Per API i dettagli, vedere [DescribePrefixLists](#) in AWS CLI Command Reference.

## describe-principal-id-format

Il seguente esempio di codice mostra come utilizzare `describe-principal-id-format`.

### AWS CLI

Per descrivere il formato ID per IAM utenti e ruoli con il formato ID lungo abilitato

L'`describe-principal-id-format` esempio seguente descrive il formato ID per l'utente root, tutti i IAM ruoli e tutti IAM gli utenti con il formato ID lungo abilitato.

```
aws ec2 describe-principal-id-format \
  --resource instance
```

Output:

```
{
  "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
    },
  ],
},
...
]
}

```

- Per API i dettagli, vedere [DescribePrincipalIdFormat](#) in AWS CLI Command Reference.

## describe-public-ipv4-pools

Il seguente esempio di codice mostra come utilizzare `describe-public-ipv4-pools`.

### AWS CLI

Per descrivere i pool di IPv4 indirizzi pubblici

L'esempio seguente mostra i dettagli sui pool di indirizzi che sono stati creati quando è stato eseguito il provisioning di intervalli di IPv4 indirizzi pubblici utilizzando Bring Your Own IP Addresses (BYOIP).

```
aws ec2 describe-public-ipv4-pools
```

Output:

```

{
  "PublicIpv4Pools": [
    {
      "PoolId": "ipv4pool-ec2-1234567890abcdef0",
      "PoolAddressRanges": [
        {
          "FirstAddress": "203.0.113.0",
          "LastAddress": "203.0.113.255",
          "AddressCount": 256,
          "AvailableAddressCount": 256
        }
      ]
    }
  ],
}

```

```
        "TotalAddressCount": 256,  
        "TotalAvailableAddressCount": 256  
      }  
    ]  
  }
```

- Per API i dettagli, vedete [DescribePublicIpv4Pools](#) in AWS CLI Command Reference.

## describe-regions

Il seguente esempio di codice mostra come utilizzare `describe-regions`.

### AWS CLI

Esempio 1: per descrivere tutte le regioni abilitate

Nell'esempio di `describe-regions` seguente vengono descritte tutte le regioni abilitate per l'account.

```
aws ec2 describe-regions
```

Output:

```
{  
  "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",
```



```
    "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"
  }
]
}
```

Per ulteriori informazioni, consulta [Regioni e zone](#) nella Amazon EC2 User Guide.

Esempio 2: per descrivere le regioni abilitate all'interno di un endpoint il cui nome contiene una stringa specifica

Nell'esempio di `describe-regions` seguente vengono descritte tutte le regioni abilitate che contengono la stringa "us" nell'endpoint.

```
aws ec2 describe-regions \
  --filters "Name=endpoint,Values=*us*"
```

Output:

```
{
  "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"
  }
]
```

Per ulteriori informazioni, consulta [Regioni e zone](#) nella Amazon EC2 User Guide.

Esempio 3: per descrivere tutte le regioni

Nell'esempio di `describe-regions` seguente vengono descritte tutte le regioni disponibili, comprese le regioni disabilite.

```
aws ec2 describe-regions \
  --all-regions
```

Output:

```
{
  "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"
}
]
}
```

Per ulteriori informazioni, consulta [Regioni e zone](#) nella Amazon EC2 User Guide.

Esempio 4: per elencare solo i nomi delle regioni

Nell'esempio di `describe-regions` seguente viene utilizzato il parametro `--query` per filtrare l'output e restituire solo i nomi delle regioni come testo.

```
aws ec2 describe-regions \  
  --all-regions \  
  --query "Regions[].{Name:RegionName}" \  
  --output text
```

Output:

```
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
```

Per ulteriori informazioni, consulta [Regioni e zone](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DescribeRegions AWS CLI Command Reference](#).

## **describe-replace-root-volume-tasks**

Il seguente esempio di codice mostra come utilizzare `describe-replace-root-volume-tasks`.



## AWS CLI

Esempio 1: per visualizzare informazioni su una specifica attività di sostituzione del volume principale

L'`describe-replace-root-volume-tasks` seguente descrive l'attività di sostituzione del volume root `replacevol-0111122223333abcd`.

```
aws ec2 describe-replace-root-volume-tasks \
  --replace-root-volume-task-ids replacevol-0111122223333abcd
```

Output:

```
{
  "ReplaceRootVolumeTasks": [
    {
      "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
      "Tags": [],
      "InstanceId": "i-0123456789abcdefa",
      "TaskState": "succeeded",
      "StartTime": "2022-03-14T15:16:28Z",
      "CompleteTime": "2022-03-14T15:16:52Z"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Sostituire un volume root](#) nella Guida per l'utente di Amazon Elastic Compute Cloud.

Esempio 2: per visualizzare informazioni su tutte le attività di sostituzione del volume root per un'istanza specifica

L'`describe-replace-root-volume-tasks` seguente descrive tutte le attività di sostituzione del volume root, ad esempio `i-0123456789abcdefa`.

```
aws ec2 describe-replace-root-volume-tasks \
  --filters Name=instance-id,Values=i-0123456789abcdefa
```

Output:

```
{
```

```
"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"  
  }  
]  
}
```

Per ulteriori informazioni, consulta [Sostituire un volume root](#) nella Guida per l'utente di Amazon Elastic Compute Cloud.

- Per API i dettagli, consulta [DescribeReplaceRootVolumeTasks AWS CLI Command Reference](#).

## describe-reserved-instances-listings

Il seguente esempio di codice mostra come utilizzare `describe-reserved-instances-listings`.

### AWS CLI

Per descrivere un elenco di istanze riservate

L'esempio seguente recupera le informazioni sull'elenco delle istanze riservate specificato.

```
aws ec2 describe-reserved-instances-listings \  
  --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DescribeReservedInstancesListings](#) in AWS CLI Command Reference.

## describe-reserved-instances-modifications

Il seguente esempio di codice mostra come utilizzare `describe-reserved-instances-modifications`.

### AWS CLI

Per descrivere le modifiche apportate alle istanze riservate

Questo comando di esempio descrive tutte le richieste di modifica delle istanze riservate che sono state inviate per il tuo account.

Comando:

```
aws ec2 describe-reserved-instances-modifications
```

Output:

```
{
  "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"
      }
    ]
  }
]
}

```

- Per API i dettagli, vedere [DescribeReservedInstancesModifications](#) in AWS CLI Command Reference.

## describe-reserved-instances-offerings

Il seguente esempio di codice mostra come utilizzare `describe-reserved-instances-offerings`.

### AWS CLI

Per descrivere le offerte di istanze riservate

Questo comando di esempio descrive tutte le istanze riservate disponibili per l'acquisto nella regione.

Comando:

```
aws ec2 describe-reserved-instances-offerings
```

Output:

```

{
  "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"
  },
  ...
}

```

Per descrivere le tue offerte di istanze riservate utilizzando le opzioni

Questo esempio elenca le istanze riservate offerte da AWS con le seguenti specifiche: tipi di istanze t1.micro, prodotti Windows (AmazonVPC) e offerte Heavy Utilization.

Comando:

```
aws ec2 describe-reserved-instances-offerings --no-include-marketplace --instance-type "t1.micro" --product-description "Windows (Amazon VPC)" --offering-type "no upfront"
```

Output:

```
{
  "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"  
  }  
]  
}
```

- Per i API dettagli, consulta Command Reference. [DescribeReservedInstancesOfferings](#) AWS CLI

## describe-reserved-instances

Il seguente esempio di codice mostra come utilizzare `describe-reserved-instances`.

### AWS CLI

Per descrivere le tue istanze riservate

Questo comando di esempio descrive le istanze riservate di cui sei proprietario.

Comando:

```
aws ec2 describe-reserved-instances
```

Output:

```
{  
  "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
  },
  ...
]
}

```

Per descrivere le istanze riservate utilizzando i filtri

Questo esempio filtra la risposta per includere solo le istanze UNIX Linux/ riservate t2.micro della durata di tre anni in us-west-1c.

Comando:

```

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

```

Output:

```

{
  "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
    }
]
}
```

Per ulteriori informazioni, consulta [Using Amazon EC2 Instances](#) nella [AWS Command Line Interface User Guide](#).

- Per API i dettagli, consulta [DescribeReservedInstances AWS CLI Command Reference](#).

## describe-route-tables

Il seguente esempio di codice mostra come utilizzare `describe-route-tables`.

### AWS CLI

Per descrivere le tabelle dei percorsi

L'esempio seguente recupera i dettagli sulle tabelle dei percorsi

```
aws ec2 describe-route-tables
```

Output:

```
{
  "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"
}
]
}

```

Per ulteriori informazioni, vedere [Working with Route Tables](#) nella Guida per l'AWS VPC utente.

- Per API i dettagli, vedere [DescribeRouteTables](#) in AWS CLI Command Reference.

## **describe-scheduled-instance-availability**

Il seguente esempio di codice mostra come utilizzare `describe-scheduled-instance-availability`.

### AWS CLI

Per descrivere una pianificazione disponibile

Questo esempio descrive una pianificazione che si verifica ogni settimana di domenica, a partire dalla data specificata.

Comando:

```
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
```

Output:

```
{  
  "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"  
    },  
    ...  
  ]  
}
```

Per restringere i risultati, è possibile aggiungere filtri che specificano il sistema operativo, la rete e il tipo di istanza.

Comando:

--filters nome=piattaforma, valori=Linux/ nome=piattaforma di rete, valori= - nome=tipo di istanza, valori=C4.large UNIX EC2 VPC

- Per API i dettagli, vedere in Command [DescribeScheduledInstanceAvailability](#) Reference AWS CLI .

## describe-scheduled-instances

Il seguente esempio di codice mostra come utilizzare `describe-scheduled-instances`.

### AWS CLI

Per descrivere le istanze pianificate

Questo esempio descrive l'istanza pianificata specificata.

Comando:

```
aws ec2 describe-scheduled-instances --scheduled-instance-ids sci-1234-1234-1234-1234-123456789012
```

Output:

```
{
  "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"
  }
]
}
```

Questo esempio descrive tutte le istanze pianificate.

Comando:

```
aws ec2 describe-scheduled-instances
```

- Per API i dettagli, consulta [DescribeScheduledInstances AWS CLI Command Reference](#).

## describe-security-group-references

Il seguente esempio di codice mostra come utilizzare `describe-security-group-references`.

AWS CLI

Per descrivere i riferimenti ai gruppi di sicurezza

Questo esempio descrive i riferimenti ai gruppi di sicurezza per `sg-bbbb2222`. La risposta indica che un gruppo `sg-bbbb2222` di sicurezza fa riferimento a un gruppo di sicurezza in VPC `vpc-aaaaaaaa`.

Comando:

```
aws ec2 describe-security-group-references --group-id sg-bbbbb22222
```

Output:

```
{
  "SecurityGroupsReferenceSet": [
    {
      "ReferencingVpcId": "vpc-aaaaaaaa ",
      "GroupId": "sg-bbbbb22222",
      "VpcPeeringConnectionId": "pcx-b04deed9"
    }
  ]
}
```

```
]
}
```

- Per API i dettagli, vedere [DescribeSecurityGroupReferences](#) in AWS CLI Command Reference.

## describe-security-group-rules

Il seguente esempio di codice mostra come utilizzare `describe-security-group-rules`.

### AWS CLI

Esempio 1: descrivere le regole del gruppo di sicurezza per un gruppo di sicurezza

L'`describe-security-group-rules` esempio seguente descrive le regole del gruppo di sicurezza di un gruppo di sicurezza specificato. Utilizzate l'`filters` opzione per assegnare i risultati a un gruppo di sicurezza specifico.

```
aws ec2 describe-security-group-rules \
  --filters Name="group-id",Values="sg-1234567890abcdef0"
```

Output:

```
{
  "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": []
    },
    {
        "SecurityGroupId": "sgr-cdef01234567890ab",
        "GroupOwnerId": "111122223333",
        "IsEgress": true,
        "IpProtocol": "-1",
        "FromPort": -1,
        "ToPort": -1,
        "CidrIpv4": "0.0.0.0/0",
        "Tags": []
    }
]
}

```

Esempio 2: descrivere una regola del gruppo di sicurezza

L'output di `aws ec2 describe-security-group-rules` seguente descrive la regola del gruppo di sicurezza specificata.

```

aws ec2 describe-security-group-rules \
  --security-group-rule-ids sgr-cdef01234567890ab

```

Output:

```

{
  "SecurityGroupRules": [
    {
      "SecurityGroupId": "sgr-cdef01234567890ab",
      "GroupOwnerId": "111122223333",
      "IsEgress": true,
      "IpProtocol": "-1",
      "FromPort": -1,
      "ToPort": -1,
      "CidrIpv4": "0.0.0.0/0",
      "Tags": []
    }
  ]
}

```



```

    }
  ]
}

```

Per ulteriori informazioni, consulta [le regole dei gruppi di sicurezza](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [DescribeSecurityGroupRules AWS CLI Command Reference](#).

## describe-security-groups

Il seguente esempio di codice mostra come utilizzare `describe-security-groups`.

### AWS CLI

Esempio 1: per descrivere un gruppo di sicurezza

Nell'esempio di `describe-security-groups` seguente viene descritto il gruppo di sicurezza specificato.

```

aws ec2 describe-security-groups \
  --group-ids sg-903004f8

```

Output:

```

{
  "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",
}
]
}

```

Esempio 2: per descrivere gruppi di sicurezza con regole specifiche

L'`describe-security-groups` esempio seguente utilizza i filtri per assegnare i risultati ai gruppi di sicurezza che dispongono di una regola che consente il SSH traffico (porta 22) e di una regola che consente il traffico proveniente da tutti gli indirizzi (`0.0.0.0/0`). Nell'esempio viene

utilizzato il parametro `--query` per visualizzare solamente i nomi dei gruppi di sicurezza. I gruppi di sicurezza devono corrispondere a tutti i filtri per essere restituiti nei risultati; tuttavia, una singola regola non deve corrispondere a tutti i filtri. Ad esempio, l'output restituisce un gruppo di sicurezza con una regola che consente il SSH traffico proveniente da un indirizzo IP specifico e un'altra regola che consente il HTTP traffico proveniente da tutti gli indirizzi.

```
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
```

Output:

```
default  
my-security-group  
web-servers  
launch-wizard-1
```

Esempio 3: per descrivere gruppi di sicurezza in base ai tag

Nell'esempio di `describe-security-groups` seguente vengono utilizzati filtri per rifinire i risultati in base ai gruppi di sicurezza che includono `test` nel nome del gruppo di sicurezza e che hanno il tag `Test=To-delete`. L'esempio utilizza il `--query` parametro per visualizzare solo i nomi e i gruppi IDs di sicurezza.

```
aws ec2 describe-security-groups \  
  --filters Name=group-name,Values=*test* Name=tag:Test,Values=To-delete \  
  --query "SecurityGroups[*].{Name:GroupName,ID:GroupId}"
```

Output:

```
[  
  {  
    "Name": "testfornewinstance",  
    "ID": "sg-33bb22aa"  
  },  
  {  
    "Name": "newgrouptest",  
    "ID": "sg-1a2b3c4d"}]
```

```
}  
]
```

Per ulteriori esempi di utilizzo dei filtri di tag, consulta [Lavorare con i tag](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DescribeSecurityGroups AWS CLI Command Reference](#).

## describe-snapshot-attribute

Il seguente esempio di codice mostra come utilizzare `describe-snapshot-attribute`.

### AWS CLI

Per descrivere gli attributi di un'istantanea

L'esempio seguente elenca gli account con cui viene condivisa un'istantanea.

```
aws ec2 describe-snapshot-attribute \  
  --snapshot-id snap-01234567890abcdef \  
  --attribute createVolumePermission
```

Output:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "CreateVolumePermissions": [  
    {  
      "UserId": "123456789012"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Share an Amazon EBS snapshot](#) nella Amazon Elastic Compute Cloud User Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeSnapshotAttribute Reference](#).

## describe-snapshot-tier-status

Il seguente esempio di codice mostra come utilizzare `describe-snapshot-tier-status`.

## AWS CLI

Per visualizzare le informazioni di archivio relative a un'istantanea archiviata

L'output di `aws ec2 describe-snapshot-tier-statuses` seguente fornisce informazioni di archiviazione su un'istantanea archiviata.

```
aws ec2 describe-snapshot-tier-status \
  --filters "Name=snapshot-id, Values=snap-01234567890abcdef"
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Visualizzazione degli snapshot archiviati](#) nella Guida per l'utente di Amazon Elastic Compute Cloud.

- Per API i dettagli, vedere [DescribeSnapshotTierStatus](#) in AWS CLI Command Reference.

## describe-snapshots

Il seguente esempio di codice mostra come utilizzare `describe-snapshots`.

### AWS CLI

Esempio 1: per descrivere uno snapshot

Nell'esempio di `describe-snapshots` seguente viene descritto lo snapshot specificato.

```
aws ec2 describe-snapshots \  
  --snapshot-ids snap-1234567890abcdef0
```

Output:

```
{  
  "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"  
        }  
      ]  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Amazon EBS snapshots](#) nella Amazon EC2 User Guide.

Esempio 2: per descrivere snapshot in base ai filtri

L'esempio seguente utilizza filtri per limitare i risultati alle istantanee di proprietà del tuo AWS account che si trovano nello stato `pending`. L'esempio utilizza il `--query` parametro per visualizzare solo l'istantanea IDs e l'ora di avvio dell'istantanea.

```
aws ec2 describe-snapshots \  
  --owner-ids self \  
  --filters Name=status,Values=pending \  
  --query "Snapshots[*].{ID:SnapshotId,Time:StartTime}"
```

## Output:

```
[
  {
    "ID": "snap-1234567890abcdef0",
    "Time": "2019-08-04T12:48:18.000Z"
  },
  {
    "ID": "snap-066877671789bd71b",
    "Time": "2019-08-04T02:45:16.000Z"
  },
  ...
]
```

Nell'esempio di `describe-snapshots` seguente vengono utilizzati filtri per rifinire i risultati in base agli snapshot creati dal volume specificato. L'esempio utilizza il `--query` parametro per visualizzare solo l'istantanea. IDs

```
aws ec2 describe-snapshots \
  --filters Name=volume-id,Values=049df61146c4d7901 \
  --query "Snapshots[*].[SnapshotId]" \
  --output text
```

## Output:

```
snap-1234567890abcdef0
snap-08637175a712c3fb9
...
```

Per ulteriori esempi di utilizzo dei filtri, consulta [Elencare e filtrare le risorse](#) nella Amazon EC2 User Guide.

Esempio 3: per descrivere snapshot in base ai tag

Nell'esempio di `describe-snapshots` seguente vengono utilizzati filtri per rifinire i risultati in base agli snapshot che hanno il tag `Stack=Prod`.

```
aws ec2 describe-snapshots \
  --filters Name=tag:Stack,Values=prod
```

Per un esempio dell'output di `describe-snapshots`, vedi l'Esempio 1.

Per ulteriori esempi di utilizzo dei filtri di tag, consulta [Lavorare con i tag](#) nella Amazon EC2 User Guide.

Esempio 4: per descrivere snapshot in base all'età

L'esempio seguente utilizza JMESPath espressioni per descrivere tutte le istantanee create dal tuo AWS account prima della data specificata. Visualizza solo l'istantaneaID.

```
aws ec2 describe-snapshots \
  --owner-ids 012345678910 \
  --query "Snapshots[?(StartTime<='2020-03-31')].[SnapshotId]"
```

Per ulteriori esempi di utilizzo dei filtri, consulta [Elencare e filtrare le risorse](#) nella Amazon EC2 User Guide.

Esempio 5: per visualizzare solo gli snapshot archiviati

Nell'esempio di `describe-snapshots` seguente vengono elencati solo gli snapshot memorizzati nel livello archivio.

```
aws ec2 describe-snapshots \
  --filters "Name=storage-tier,Values=archive"
```

Output:

```
{
  "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": []
      },
    ]
  }
```

Per ulteriori informazioni, consulta [Visualizzazione degli snapshot archiviati](#) nella Guida per l'utente di Amazon Elastic Compute Cloud.

- Per API i dettagli, consulta [DescribeSnapshots AWS CLI Command Reference](#).

## **describe-spot-datafeed-subscription**

Il seguente esempio di codice mostra come utilizzare `describe-spot-datafeed-subscription`.

AWS CLI

Per descrivere l'abbonamento al datafeed di Spot Instance per un account

Questo comando di esempio descrive il feed di dati per l'account.

Comando:

```
aws ec2 describe-spot-datafeed-subscription
```

Output:

```
{
  "SpotDatafeedSubscription": {
    "OwnerId": "123456789012",
    "Prefix": "spotdata",
    "Bucket": "my-s3-bucket",
    "State": "Active"
  }
}
```

- Per API i dettagli, vedere [DescribeSpotDatafeedSubscription](#) in AWS CLI Command Reference.

## **describe-spot-fleet-instances**

Il seguente esempio di codice mostra come utilizzare `describe-spot-fleet-instances`.

## AWS CLI

Per descrivere le istanze Spot associate a un parco istanze Spot

Questo comando di esempio elenca le istanze Spot associate al parco istanze Spot specificato.

Comando:

```
aws ec2 describe-spot-fleet-instances --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Output:

```
{
  "ActiveInstances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "InstanceType": "m3.medium",
      "SpotInstanceRequestId": "sir-08b93456"
    },
    ...
  ],
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

- Per API i dettagli, vedere [DescribeSpotFleetInstances](#) in AWS CLI Command Reference.

## **describe-spot-fleet-request-history**

Il seguente esempio di codice mostra come utilizzare `describe-spot-fleet-request-history`.

## AWS CLI

Per descrivere la storia della flotta Spot

Questo comando di esempio restituisce la cronologia della flotta Spot specificata a partire dall'ora specificata.

Comando:

```
aws ec2 describe-spot-fleet-request-history --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --start-time 2015-05-26T00:00:00Z
```

L'output di esempio seguente mostra i lanci riusciti di due istanze Spot per il parco istanze Spot.

Output:

```
{
  "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": "CpHNsscimcV5oH7bSsub03CI2Qms5+ypNpNm
+53MNlR0YcXAkp0xF1fKf91yVxSExmbtma3awYxMFzNA663ZskT0AhtJ6TCb2Z8bQC2EnZgyELbymtWPfpZ1ZbauVg
+P+TfG1WxWWB/Vr5dk5d4LfdgA/DRAHUrYgxzrEXAMPLE=",
  "StartTime": "2015-05-26T00:00:00Z"
}
```

- Per API i dettagli, consulta AWS CLI Command [DescribeSpotFleetRequestHistory](#) Reference.

## describe-spot-fleet-requests

Il seguente esempio di codice mostra come utilizzare `describe-spot-fleet-requests`.

### AWS CLI

Per descrivere le richieste della tua flotta Spot

Questo esempio descrive tutte le richieste della tua flotta Spot.

Comando:

```
aws ec2 describe-spot-fleet-requests
```

Output:

```
{
  "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"
}
]
}
```

Per descrivere una richiesta relativa alla flotta Spot

Questo esempio descrive la richiesta di flotta Spot specificata.

Comando:

```
aws ec2 describe-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Output:

```
{
  "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"
}
]
}

```

- Per API i dettagli, vedere [DescribeSpotFleetRequests](#) in AWS CLI Command Reference.

## describe-spot-instance-requests

Il seguente esempio di codice mostra come utilizzare `describe-spot-instance-requests`.

### AWS CLI

Esempio 1: per descrivere una richiesta di istanza Spot

L'`describe-spot-instance-requests` esempio seguente descrive la richiesta di istanza Spot specificata.

```
aws ec2 describe-spot-instance-requests \
  --spot-instance-request-ids sir-08b93456
```

Output:

```

{
  "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"
}
```



```
]
}
```

Esempio 2: Per descrivere le richieste di istanze Spot in base ai filtri

L'`describe-spot-instance-requests` seguente utilizza i filtri per assegnare i risultati alle richieste di istanze Spot con il tipo di istanza specificato nella zona di disponibilità specificata. L'esempio utilizza il `--query` parametro per visualizzare solo l'istanzaID.

```
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
```

Output:

```
i-057750d42936e468a
i-001efd250faaa6ffa
i-027552a73f021f3bd
...
```

Per ulteriori esempi di utilizzo dei filtri, consulta [Elencare e filtrare le risorse](#) nella Amazon Elastic Compute Cloud User Guide.

Esempio 3: per descrivere le richieste di istanze Spot in base ai tag

L'`describe-spot-instance-requests` seguente utilizza i filtri di tag per indirizzare i risultati alle richieste di istanze Spot che contengono il tag `cost-center=cc123`.

```
aws ec2 describe-spot-instance-requests \
  --filters Name=tag:cost-center,Values=cc123
```

Per un esempio dell'output di `describe-spot-instance-requests`, vedi l'Esempio 1.

Per ulteriori esempi di utilizzo dei filtri di tag, consulta [Lavorare con i tag](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DescribeSpotInstanceRequests AWS CLI](#) Command Reference.

## describe-spot-price-history

Il seguente esempio di codice mostra come utilizzare `describe-spot-price-history`.

### AWS CLI

Per descrivere la cronologia dei prezzi Spot

Questo comando di esempio restituisce la cronologia dei prezzi Spot per le istanze `m1.xlarge` per un particolare giorno di gennaio.

Comando:

```
aws ec2 describe-spot-price-history --instance-types m1.xlarge --start-time 2014-01-06T07:08:09 --end-time 2014-01-06T08:09:10
```

Output:

```
{
  "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"
    },
    ...
  ]
}
```

Per descrivere la cronologia dei prezzi Spot per Linux/Amazon UNIX VPC

Questo comando di esempio restituisce la cronologia dei prezzi Spot per le VPC istanze m1.xlarge, UNIX Linux/Amazon per un particolare giorno di gennaio.

Comando:

```
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
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

- Per API i dettagli, consulta Command Reference. [DescribeSpotPriceHistory](#) AWS CLI

## describe-stale-security-groups

Il seguente esempio di codice mostra come utilizzare `describe-stale-security-groups`.

AWS CLI

Per descrivere i gruppi di sicurezza obsoleti

Questo esempio descrive le regole obsolete dei gruppi di sicurezza per. `vpc-11223344` La risposta mostra che `sg-5fa68d3a` nel tuo account ha una regola di ingresso obsoleta a cui fa riferimento `sg-279ab042` nel peer e che `sg-fe6fba9a` nel tuo account ha una SSH regola di uscita obsoleta che fa riferimento al peerVPC. SSH `sg-ef6fba8b` VPC

Comando:

```
aws ec2 describe-stale-security-groups --vpc-id vpc-11223344
```

Output:

```
{
  "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"
}
]
}

```

- [API DescribeStaleSecurityGroups](#) Per i dettagli AWS CLI , consulta Command Reference.

## describe-store-image-tasks

Il seguente esempio di codice mostra come utilizzare `describe-store-image-tasks`.

### AWS CLI

Per descrivere lo stato di avanzamento di un'attività in AMI negozio

L'`describe-store-image-tasks` esempio seguente descrive lo stato di avanzamento di un'attività di AMI archiviazione.

```
aws ec2 describe-store-image-tasks
```

Output:

```

{
  "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"
    }
]
}
```

Per ulteriori informazioni sull'archiviazione e il ripristino di un dispositivo AMI utilizzando S3, consulta [Archiviare e ripristinare un AMI utilizzo di 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](https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/ami-store-restore.html)> nella Amazon User Guide. EC2

- Per API [DescribeStoreImageTasks](#) AWS CLI dettagli, consulta Command Reference.

## describe-subnets

Il seguente esempio di codice mostra come utilizzare `describe-subnets`.

### AWS CLI

Esempio 1: per descrivere tutte le sottoreti

Nell'esempio di `describe-subnets` seguente vengono visualizzati i dettagli delle sottoreti.

```
aws ec2 describe-subnets
```

Output:

```
{
  "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
    }
  }
]
}

```

Per ulteriori informazioni, vedere [Working with VPCs and Subnet nella Guida](#) per l'AWS VPCutente.

Esempio 2: Per descrivere le sottoreti di una specifica VPC

L'`describe-subnets`esempio seguente utilizza un filtro per recuperare i dettagli per le sottoreti di quanto specificato. VPC

```
aws ec2 describe-subnets \  
  --filters "Name=vpc-id,Values=vpc-3EXAMPLE"
```

Output:

```
{  
  "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  
      }  
    }  
  ]  
}
```



```

    }
  }
]
}

```

Per ulteriori informazioni, vedere [Working with VPCs and Subnet](#) nella Guida per l'utente AWS VPC.

Esempio 3: per descrivere le sottoreti con un tag specifico

L'esempio seguente utilizza un filtro per recuperare i dettagli di quelle sottoreti con il tag `CostCenter=123` e il `--query` parametro per visualizzare la sottorete IDs delle sottoreti con questo tag.

```

aws ec2 describe-subnets \
  --filters "Name=tag:CostCenter,Values=123" \
  --query "Subnets[*].SubnetId" \
  --output text

```

Output:

```

subnet-0987a87c8b37348ef
subnet-02a95061c45f372ee
subnet-03f720e7de2788d73

```

Per ulteriori informazioni, consulta [Working with VPCs and Subnet](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeSubnets](#) Reference.

## describe-tags

Il seguente esempio di codice mostra come utilizzare `describe-tags`.

AWS CLI

Esempio 1: Per descrivere tutti i tag per una singola risorsa

L'esempio seguente descrive i tag per l'istanza specificata.

```

aws ec2 describe-tags \
  --filters "Name=resource-id,Values=i-1234567890abcdef8"

```

## Output:

```
{
  "Tags": [
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef8",
      "Value": "Test",
      "Key": "Stack"
    },
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef8",
      "Value": "Beta Server",
      "Key": "Name"
    }
  ]
}
```

Esempio 2: Per descrivere tutti i tag per un tipo di risorsa

L'`describe-tags` seguente descrive i tag per i tuoi volumi.

```
aws ec2 describe-tags \
  --filters "Name=resource-type,Values=volume"
```

## Output:

```
{
  "Tags": [
    {
      "ResourceType": "volume",
      "ResourceId": "vol-1234567890abcdef0",
      "Value": "Project1",
      "Key": "Purpose"
    },
    {
      "ResourceType": "volume",
      "ResourceId": "vol-049df61146c4d7901",
      "Value": "Logs",
      "Key": "Purpose"
    }
  ]
}
```

```
}
```

Esempio 3: per descrivere tutti i tag

L'`describe-tags` seguente descrive i tag per tutte le tue risorse.

```
aws ec2 describe-tags
```

Esempio 4: Per descrivere i tag delle risorse in base a una chiave di tag

L'`describe-tags` seguente descrive i tag per le tue risorse che hanno un tag con la chiave `Stack`.

```
aws ec2 describe-tags \  
  --filters Name=key,Values=Stack
```

Output:

```
{  
  "Tags": [  
    {  
      "ResourceType": "volume",  
      "ResourceId": "vol-027552a73f021f3b",  
      "Value": "Production",  
      "Key": "Stack"  
    },  
    {  
      "ResourceType": "instance",  
      "ResourceId": "i-1234567890abcdef8",  
      "Value": "Test",  
      "Key": "Stack"  
    }  
  ]  
}
```

Esempio 5: Per descrivere i tag delle risorse in base a una chiave di tag e a un valore del tag

L'`describe-tags` seguente descrive i tag delle risorse che hanno il tag `Stack=Test`.

```
aws ec2 describe-tags \  
  --filters Name=key,Values=Stack Name=value,Values=Test
```

## Output:

```
{
  "Tags": [
    {
      "ResourceType": "image",
      "ResourceId": "ami-3ac336533f021f3bd",
      "Value": "Test",
      "Key": "Stack"
    },
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef8",
      "Value": "Test",
      "Key": "Stack"
    }
  ]
}
```

L'`describe-tags`esempio seguente utilizza una sintassi alternativa per descrivere le risorse con il tag. `Stack=Test`

```
aws ec2 describe-tags \
  --filters "Name=tag:Stack,Values=Test"
```

L'`describe-tags`esempio seguente descrive i tag per tutte le istanze che hanno un tag con la chiave `Purpose` e nessun valore.

```
aws ec2 describe-tags \
  --filters "Name=resource-
  type,Values=instance" "Name=key,Values=Purpose" "Name=value,Values="
```

## Output:

```
{
  "Tags": [
    {
      "ResourceType": "instance",
      "ResourceId": "i-1234567890abcdef5",
      "Value": null,
      "Key": "Purpose"
    }
  ]
}
```

```
]
}
```

- Per API i dettagli, consultate [DescribeTags AWS CLI Command Reference](#).

## describe-traffic-mirror-filters

Il seguente esempio di codice mostra come utilizzare `describe-traffic-mirror-filters`.

### AWS CLI

Per visualizzare i filtri del Traffic Mirror

L'esempio seguente mostra i dettagli di tutti i filtri Traffic Mirror.

```
aws ec2 describe-traffic-mirror-filters
```

Output:

```
{
  "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": []
    }
  ]
}
```

```
]
}
```

Per ulteriori informazioni, consulta [Visualizza i filtri del Traffic Mirroring](#) nella Traffic Mirroring Guide.

- Per API i dettagli, consulta [DescribeTrafficMirrorFilters AWS CLI Command Reference](#).

## **describe-traffic-mirror-sessions**

Il seguente esempio di codice mostra come utilizzare `describe-traffic-mirror-sessions`.

### AWS CLI

Per descrivere una sessione di Traffic Mirror

L'esempio seguente mostra i dettagli delle sessioni di Traffic Mirror.

```
aws ec2 describe-traffic-mirror-sessions
```

Output:

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Visualizza i dettagli della sessione di Traffic Mirror](#) nella AWS Traffic Mirroring Guide.

- Per API i dettagli, consulta [DescribeTrafficMirrorSessions AWS CLI Command Reference](#).

## describe-traffic-mirror-targets

Il seguente esempio di codice mostra come utilizzare `describe-traffic-mirror-targets`.

### AWS CLI

Descrivere un obiettivo speculare

L'esempio seguente visualizza informazioni sul target del Traffic Mirror specificato.

```

aws ec2 describe-traffic-mirror-targets \
  --traffic-mirror-target-ids tmt-0dabe9b0a6EXAMPLE

```

Output:

```

{
  "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": []
  }
]
}

```

Per ulteriori informazioni, consulta [Traffic Mirroring targets](#) nella Amazon VPC Traffic Mirroring Guide.

- Per API i dettagli, consulta [DescribeTrafficMirrorTargets AWS CLI Command Reference](#).

## describe-transit-gateway-attachments

Il seguente esempio di codice mostra come utilizzare `describe-transit-gateway-attachments`.

### AWS CLI

Per visualizzare gli allegati del gateway di transito

L'esempio seguente mostra i dettagli degli allegati del gateway di transito.

```
aws ec2 describe-transit-gateway-attachments
```

Output:

```

{
  "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": []
}
]
}

```

Per ulteriori informazioni, consulta [Work with Transit Gateways](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [DescribeTransitGatewayAttachments](#) in AWS CLI Command Reference.

## describe-transit-gateway-connect-peers

Il seguente esempio di codice mostra come utilizzare `describe-transit-gateway-connect-peers`.

### AWS CLI

Per descrivere un peer Transit Gateway Connect

L'esempio seguente descrive il peer Connect specificato.

```

aws ec2 describe-transit-gateway-connect-peers \
  --transit-gateway-connect-peer-ids tgw-connect-peer-0666adbac4EXAMPLE

```

Output:

```

{
  "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": []
}
]
}
}

```

Per ulteriori informazioni, consulta [gli allegati Transit gateway Connect e i peer Transit Gateway Connect](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [DescribeTransitGatewayConnectPeers](#) in AWS CLI Command Reference.

## describe-transit-gateway-connects

Il seguente esempio di codice mostra come utilizzare `describe-transit-gateway-connects`.

### AWS CLI

Per descrivere un gateway di transito (allegato Connect)

L'esempio seguente descrive l'allegato Connect specificato.

```
aws ec2 describe-transit-gateway-connects \
```

```
--transit-gateway-attachment-ids tgw-attach-037012e5dcEXAMPLE
```

Output:

```
{
  "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": []
    }
  ]
}
```

Per ulteriori informazioni, consulta [gli allegati Transit gateway Connect e i peer Transit Gateway Connect](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [DescribeTransitGatewayConnects](#) in AWS CLI Command Reference.

## describe-transit-gateway-multicast-domains

Il seguente esempio di codice mostra come utilizzare `describe-transit-gateway-multicast-domains`.

AWS CLI

Per descrivere i domini multicast del gateway di transito

L'`describe-transit-gateway-multicast-domains` esempio seguente mostra i dettagli per tutti i domini multicast del gateway di transito.

```
aws ec2 describe-transit-gateway-multicast-domains
```

Output:

```
{
```

```
"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"
      }
    ]
  }
]
```

Per ulteriori informazioni, consulta [Managing multicast domain](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [DescribeTransitGatewayMulticastDomains](#) in AWS CLI Command Reference.

## **describe-transit-gateway-peering-attachments**

Il seguente esempio di codice mostra come utilizzare `describe-transit-gateway-peering-attachments`.

### AWS CLI

Per descrivere gli allegati di peering del tuo Transit Gateway

L'esempio seguente mostra i dettagli di tutti gli allegati di peering del gateway Transit.

```
aws ec2 describe-transit-gateway-peering-attachments
```

Output:

```
{
  "TransitGatewayPeeringAttachments": [
    {
      "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": "pendingAcceptance",
      "CreationTime": "2019-12-09T11:38:05.000Z",
      "Tags": []
    }
  ]
}
```

Per ulteriori informazioni, consulta [Transit Gateway Peering Attachments](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [DescribeTransitGatewayPeeringAttachments](#) in AWS CLI Command Reference.

## describe-transit-gateway-policy-tables

Il seguente esempio di codice mostra come utilizzare `describe-transit-gateway-policy-tables`.

AWS CLI

Per descrivere una tabella di policy sui gateway di transito

L'`describe-transit-gateway-policy-tables` esempio seguente descrive la tabella delle politiche del gateway di transito specificata.

```
aws ec2 describe-transit-gateway-policy-tables \
  --transit-gateway-policy-table-ids tgw-ptb-0a16f134b78668a81
```

Output:

```
{
  "TransitGatewayPolicyTables": [
    {
      "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
      "TransitGatewayId": "tgw-067f8505c18f0bd6e",
      "State": "available",
      "CreationTime": "2023-11-28T16:36:43+00:00",
      "Tags": []
    }
  ]
}
```

Per ulteriori informazioni, consulta le [tabelle delle policy del Transit Gateway](#) nella Transit Gateway User Guide.

- Per API i dettagli, vedere [DescribeTransitGatewayPolicyTables](#) in AWS CLI Command Reference.

## describe-transit-gateway-route-tables

Il seguente esempio di codice mostra come utilizzare `describe-transit-gateway-route-tables`.

AWS CLI

Per descrivere le tabelle dei percorsi dei gateway di transito

L'esempio seguente mostra i dettagli delle tabelle degli itinerari dei gateway di transito.

```
aws ec2 describe-transit-gateway-route-tables
```

Output:

```
{
  "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": []
    }
  ]
}
```

Per ulteriori informazioni, consulta [Visualizza le tabelle degli itinerari dei gateway di transito](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [DescribeTransitGatewayRouteTables](#) in AWS CLI Command Reference.

## describe-transit-gateway-vpc-attachments

Il seguente esempio di codice mostra come utilizzare `describe-transit-gateway-vpc-attachments`.

AWS CLI

Per descrivere gli VPC allegati del gateway di transito

L'`describe-transit-gateway-vpc-attachment` esempio seguente mostra i dettagli degli VPC allegati del gateway di transito.

```
aws ec2 describe-transit-gateway-vpc-attachments
```

Output:



```
{
  "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"
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, consulta [Visualizza VPC gli allegati nella](#) Transit Gateways Guide.

- Per API i dettagli, vedere [DescribeTransitGatewayVpcAttachments](#) in AWS CLI Command Reference.

## **describe-transit-gateways**

Il seguente esempio di codice mostra come utilizzare `describe-transit-gateways`.

### AWS CLI

Per descrivere i gateway di transito

L'`describe-transit-gateways` esempio seguente recupera i dettagli sui gateway di transito.

**aws ec2 describe-transit-gateways**

## Output:

```
{
  "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"
      }
    ]
  }
]
}

```

- Per API i dettagli, vedere [DescribeTransitGateways](#) in AWS CLI Command Reference.

## describe-verified-access-endpoints

Il seguente esempio di codice mostra come utilizzare `describe-verified-access-endpoints`.

### AWS CLI

Per descrivere un endpoint di accesso verificato

L'`delete-verified-access-endpoint` esempio seguente descrive l'endpoint di accesso verificato specificato.

```

aws ec2 describe-verified-access-endpoints \
  --verified-access-endpoint-ids vae-066fac616d4d546f2

```

Output:

```

{
  "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"
      }
    ]
  }
]
}

```

Per ulteriori informazioni, consulta gli [endpoint di accesso verificato nella Guida](#) per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [DescribeVerifiedAccessEndpoints AWS CLI Command Reference](#).

## describe-verified-access-groups

Il seguente esempio di codice mostra come utilizzare `describe-verified-access-groups`.

### AWS CLI

Per descrivere un gruppo con accesso verificato

L'esempio seguente descrive il gruppo di accesso verificato specificato.

```

aws ec2 describe-verified-access-groups \
  --verified-access-group-ids vagr-0dbe967baf14b7235

```

**Output:**

```
{
  "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"
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, consulta [Gruppi di accesso verificato](#) nella Guida per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [DescribeVerifiedAccessGroups AWS CLI Command Reference](#).

**describe-verified-access-instance-logging-configurations**

Il seguente esempio di codice mostra come utilizzare `describe-verified-access-instance-logging-configurations`.

**AWS CLI**

Per descrivere la configurazione di registrazione per un'istanza di accesso verificato

L'`describe-verified-access-instance-logging-configurations` esempio seguente descrive la configurazione di registrazione per l'istanza di accesso verificato specificata.

```
aws ec2 describe-verified-access-instance-logging-configurations \
  --verified-access-instance-ids vai-0ce000c0b7643abea
```

**Output:**

```
{
  "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
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta [i registri di accesso verificato nella Guida](#) per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [DescribeVerifiedAccessInstanceLoggingConfigurations AWS CLI Command Reference](#).

**describe-verified-access-instances**

Il seguente esempio di codice mostra come utilizzare `describe-verified-access-instances`.

**AWS CLI**

Per descrivere un'istanza di accesso verificato

L'`describe-verified-access-instances`esempio seguente descrive l'istanza di accesso verificato specificata.

```
aws ec2 describe-verified-access-instances \
  --verified-access-instance-ids vai-0ce000c0b7643abea
```

Output:

```
{
  "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"
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, consulta le [istanze di accesso verificato](#) nella Guida per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [DescribeVerifiedAccessInstances AWS CLI Command Reference](#).

## describe-verified-access-trust-providers

Il seguente esempio di codice mostra come utilizzare `describe-verified-access-trust-providers`.

AWS CLI

Per descrivere un provider fiduciario Verified Access

L'output di `aws ec2 describe-verified-access-trust-providers` seguente descrive il provider fiduciario Verified Access specificato.

```
aws ec2 describe-verified-access-trust-providers \
  --verified-access-trust-provider-ids vatp-0bb32de759a3e19e7
```

Output:

```
{
  "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"
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, consulta [Trust providers for Verified Access](#) nella AWS Verified Access User Guide.

- Per API i dettagli, consulta [DescribeVerifiedAccessTrustProviders AWS CLI Command Reference](#).

## describe-volume-attribute

Il seguente esempio di codice mostra come utilizzare `describe-volume-attribute`.

### AWS CLI

Per descrivere un attributo di volume



Questo comando di esempio descrive l'autoEnableIoattributo del volume con l'IDvol-049df61146c4d7901.

Comando:

```
aws ec2 describe-volume-attribute --volume-id vol-049df61146c4d7901 --  
attribute autoEnableIO
```

Output:

```
{  
  "AutoEnableIO": {  
    "Value": false  
  },  
  "VolumeId": "vol-049df61146c4d7901"  
}
```

- Per API i dettagli, vedere [DescribeVolumeAttribute](#) in AWS CLI Command Reference.

## describe-volume-status

Il seguente esempio di codice mostra come utilizzare `describe-volume-status`.

AWS CLI

Per descrivere lo stato di un singolo volume

Questo comando di esempio descrive lo stato del volume `vol-1234567890abcdef0`.

Comando:

```
aws ec2 describe-volume-status --volume-ids vol-1234567890abcdef0
```

Output:

```
{  
  "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": []
}
]
}

```

Per descrivere lo stato dei volumi danneggiati

Questo comando di esempio descrive lo stato di tutti i volumi danneggiati. In questo output di esempio, non ci sono volumi danneggiati.

Comando:

```
aws ec2 describe-volume-status --filters Name=volume-status.status,Values=impaired
```

Output:

```
{
  "VolumeStatuses": []
}
```

Se hai un volume con un controllo dello stato non riuscito (lo stato è compromesso), consulta [Working with an Impaired Volume](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeVolumeStatus](#)Reference.

## describe-volumes-modifications

Il seguente esempio di codice mostra come utilizzare `describe-volumes-modifications`.

## AWS CLI

Per descrivere lo stato di modifica di un volume

L'`describe-volumes-modifications` seguente descrive lo stato di modifica del volume specificato.

```
aws ec2 describe-volumes-modifications \  
  --volume-ids vol-1234567890abcdef0
```

Output:

```
{  
  "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  
  }  
}
```

- Per API i dettagli, vedere [DescribeVolumesModifications](#) in AWS CLI Command Reference.

## `describe-volumes`

Il seguente esempio di codice mostra come utilizzare `describe-volumes`.

### AWS CLI

Esempio 1: descrivere un volume

L'`describe-volumes` seguente descrive i volumi specificati nella regione corrente.

```
aws ec2 describe-volumes \  
  --volume-ids vol-049df61146c4d7901 vol-1234567890abcdef0
```

## Output:

```
{
  "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
    }
  ]
}
```

Esempio 2: Per descrivere i volumi collegati a un'istanza specifica

L'`describe-volumes` seguente descrive tutti i volumi che sono entrambi collegati all'istanza specificata e impostati per essere eliminati quando l'istanza termina.

```
aws ec2 describe-volumes \  
  --region us-east-1 \  
  --filters Name=attachment.instance-  
id,Values=i-1234567890abcdef0 Name=attachment.delete-on-termination,Values=true
```

Per un esempio dell'output di `describe-volumes`, vedi l'Esempio 1.

Esempio 3: descrivere i volumi disponibili in una zona di disponibilità specifica

L'`describe-volumes` seguente descrive tutti i volumi che hanno lo stato di `available` e si trovano nella zona di disponibilità specificata.

```
aws ec2 describe-volumes \  
  --filters Name=status,Values=available Name=availability-zone,Values=us-east-1a
```

Per un esempio dell'output di `describe-volumes`, vedi l'Esempio 1.

Esempio 4: descrivere i volumi in base ai tag

L'`describe-volumes` seguente descrive tutti i volumi che hanno la chiave del tag `Name` e un valore che inizia con `Test`. L'output viene quindi filtrato con una query che visualizza solo i tag e ID i volumi.

```
aws ec2 describe-volumes \  
  --filters Name=tag:Name,Values=Test* \  
  --query "Volumes[*].{ID:VolumeId,Tag:Tags}"
```

Output:

```
[  
  {  
    "Tag": [  
      {  
        "Value": "Test2",  
        "Key": "Name"  
      }  
    ],  
    "ID": "vol-1234567890abcdef0"
```

```
    },
    {
      "Tag": [
        {
          "Value": "Test1",
          "Key": "Name"
        }
      ],
      "ID": "vol-049df61146c4d7901"
    }
  ]
}
```

Per ulteriori esempi di utilizzo dei filtri di tag, consulta [Lavorare con i tag](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DescribeVolumes AWS CLI Command Reference](#).

## describe-vpc-attribute

Il seguente esempio di codice mostra come utilizzare `describe-vpc-attribute`.

### AWS CLI

Per descrivere l' `enableDnsSupport` attributo

Questo esempio descrive l'`enableDnsSupport` attributo. Questo attributo indica se DNS la risoluzione è abilitata per VPC. Se questo attributo è `true`, il DNS server Amazon risolve i DNS nomi host delle tue istanze negli indirizzi IP corrispondenti; in caso contrario, non lo fa.

Comando:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsSupport
```

Output:

```
{
  "VpcId": "vpc-a01106c2",
  "EnableDnsSupport": {
    "Value": true
  }
}
```

Per descrivere l'attributo `enableDnsHostnames`

Questo esempio descrive l'attributo `enableDnsHostnames`. Questo attributo indica se le istanze avviate in VPC get DNS hostnames. Se questo attributo è `true`, le istanze contenute in VPC get DNS hostnames; in caso contrario, non lo fanno.

Comando:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsHostnames
```

Output:

```
{
  "VpcId": "vpc-a01106c2",
  "EnableDnsHostnames": {
    "Value": true
  }
}
```

- Per API i dettagli, vedere [DescribeVpcAttribute](#) in AWS CLI Command Reference.

## **describe-vpc-classic-link-dns-support**

Il seguente esempio di codice mostra come utilizzare `describe-vpc-classic-link-dns-support`.

AWS CLI

Per descrivere ClassicLink DNS il supporto per VPCs

Questo esempio descrive lo stato del ClassicLink DNS supporto di tutti i tuoi VPCs.

Comando:

```
aws ec2 describe-vpc-classic-link-dns-support
```

Output:

```
{
  "Vpcs": [
```

```
{
  "VpcId": "vpc-88888888",
  "ClassicLinkDnsSupported": true
},
{
  "VpcId": "vpc-1a2b3c4d",
  "ClassicLinkDnsSupported": false
}
]
```

- Per API i dettagli, vedere [DescribeVpcClassicLinkDnsSupport](#) in AWS CLI Command Reference.

## describe-vpc-classic-link

Il seguente esempio di codice mostra come utilizzare `describe-vpc-classic-link`.

### AWS CLI

Per descrivere lo ClassicLink stato del tuo VPCs

Questo esempio elenca lo ClassicLink stato di `vpc-88888888`.

Comando:

```
aws ec2 describe-vpc-classic-link --vpc-id vpc-88888888
```

Output:

```
{
  "Vpcs": [
    {
      "ClassicLinkEnabled": true,
      "VpcId": "vpc-88888888",
      "Tags": [
        {
          "Value": "classiclinkvpc",
          "Key": "Name"
        }
      ]
    }
  ]
}
```



```
]
}
```

Questo esempio elenca solo VPCs quelli abilitati per Classiclink (il valore del `is-classic-link-enabled` filtro è impostato su). `true`

Comando:

```
aws ec2 describe-vpc-classic-link --filter "Name=is-classic-link-
enabled,Values=true"
```

- Per API i dettagli, vedere [DescribeVpcClassicLink](#) in AWS CLI Command Reference.

## describe-vpc-endpoint-connection-notifications

Il seguente esempio di codice mostra come utilizzare `describe-vpc-endpoint-connection-notifications`.

AWS CLI

Per descrivere le notifiche di connessione agli endpoint

L'`describe-vpc-endpoint-connection-notifications` esempio seguente descrive tutte le notifiche di connessione all'endpoint.

```
aws ec2 describe-vpc-endpoint-connection-notifications
```

Output:

```
{
  "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"
  }
]
}

```

- Per API i dettagli, vedere [DescribeVpcEndpointConnections](#) in AWS CLI Command Reference.

## describe-vpc-endpoint-connections

Il seguente esempio di codice mostra come utilizzare `describe-vpc-endpoint-connections`.

### AWS CLI

Per descrivere le connessioni agli VPC endpoint

Questo esempio descrive le connessioni degli endpoint dell'interfaccia al servizio endpoint e filtra i risultati per visualizzare gli endpoint che lo sono. `PendingAcceptance`

Comando:

```
aws ec2 describe-vpc-endpoint-connections --filters Name=vpc-endpoint-
state,Values=pendingAcceptance
```

Output:

```

{
  "VpcEndpointConnections": [
    {
      "VpcEndpointId": "vpce-0abed31004e618123",
      "ServiceId": "vpce-svc-0abced088d20def56",
      "CreationTimestamp": "2017-11-30T10:00:24.350Z",
      "VpcEndpointState": "pendingAcceptance",
      "VpcEndpointOwner": "123456789012"
    }
  ]
}

```

- Per API i dettagli, vedere [DescribeVpcEndpointConnections](#) in AWS CLI Command Reference.

## describe-vpc-endpoint-service-configurations

Il seguente esempio di codice mostra come utilizzare `describe-vpc-endpoint-service-configurations`.

### AWS CLI

Per descrivere le configurazioni dei servizi endpoint

L'`describe-vpc-endpoint-service-configurations`esempio seguente descrive le configurazioni del servizio endpoint.

```
aws ec2 describe-vpc-endpoint-service-configurations
```

Output:

```
{
  "ServiceConfigurations": [
    {
      "ServiceType": [
        {
          "ServiceType": "GatewayLoadBalancer"
        }
      ],
      "ServiceId": "vpce-svc-012d33a1c4321cab",
      "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-012d33a1c4321cab",
      "ServiceState": "Available",
      "AvailabilityZones": [
        "us-east-1d"
      ],
      "AcceptanceRequired": false,
      "ManagesVpcEndpoints": false,
      "GatewayLoadBalancerArns": [
        "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/gwy/GWLBService/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": []
  }
]
}
```

Per ulteriori informazioni, consulta [i servizi VPC endpoint](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [DescribeVpcEndpointServiceConfigurations AWS CLI Command Reference](#).

## **describe-vpc-endpoint-service-permissions**

Il seguente esempio di codice mostra come utilizzare `describe-vpc-endpoint-service-permissions`.

### AWS CLI

Per descrivere le autorizzazioni del servizio endpoint

Questo esempio descrive le autorizzazioni per il servizio endpoint specificato.

Comando:

```
aws ec2 describe-vpc-endpoint-service-permissions --service-id vpce-  
svc-03d5ebb7d9579a2b3
```

Output:

```
{  
  "AllowedPrincipals": [  
    {  
      "PrincipalType": "Account",  
      "Principal": "arn:aws:iam::123456789012:root"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [DescribeVpcEndpointServicePermissions](#) in AWS CLI Command Reference.

## describe-vpc-endpoint-services

Il seguente esempio di codice mostra come utilizzare `describe-vpc-endpoint-services`.

AWS CLI

Esempio 1: Per descrivere tutti i VPC servizi endpoint

Il seguente esempio "describe-vpc-endpoint-services" elenca tutti i servizi VPC endpoint per una AWS regione.

```
aws ec2 describe-vpc-endpoint-services
```

Output:

```
{  
  "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"
]
}

```

Per ulteriori informazioni, consulta [Visualizza i nomi dei AWS servizi disponibili](#) nella Guida per AWS PrivateLink l'utente di.

Esempio 2: Per descrivere i dettagli su un servizio endpoint

Il seguente esempio "describe-vpc-endpoint-services" elenca i dettagli del servizio endpoint dell'interfaccia Amazon S3

```

aws ec2 describe-vpc-endpoint-services \
  --filter "Name=service-type,Values=Interface" Name=service-
  name,Values=com.amazonaws.us-east-1.s3

```

**Output:**

```
{
  "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"
  ]
}
```

Per ulteriori informazioni, consulta [Visualizza i nomi di AWS servizio disponibili nella Guida per l'utente per AWS PrivateLink](#)

- Per API i dettagli, vedere [DescribeVpcEndpointServices](#) in AWS CLI Command Reference.

**describe-vpc-endpoints**

Il seguente esempio di codice mostra come utilizzare `describe-vpc-endpoints`.



## AWS CLI

Per descrivere i tuoi VPC endpoint

L'`describe-vpc-endpoints` seguente mostra i dettagli di tutti gli VPC endpoint.

```
aws ec2 describe-vpc-endpoints
```

Output:

```
{
  "VpcEndpoints": [
    {
      "PolicyDocument": "{\n  \"Version\": \"2008-10-17\",\n  \"Statement\": [\n    {\n      \"Effect\": \"Allow\",\n      \"Principal\": \"*\",\n      \"Action\": \"*\",\n      \"Resource\": \"*\"\n    }\n  ]\n}",
      "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  \"Version\": \"2008-10-17\",\n  \"Statement\": [\n    {\n      \"Effect\": \"Allow\",\n      \"Principal\": \"*\",\n      \"Action\": \"*\",\n      \"Resource\": \"*\"\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"
  }
]
```

Per ulteriori informazioni, consulta gli [VPCendpoint](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [DescribeVpcEndpoints AWS CLI Command Reference](#).

## describe-vpc-peering-connections

Il seguente esempio di codice mostra come utilizzare `describe-vpc-peering-connections`.

### AWS CLI

Per descrivere le connessioni di VPC peering

Questo esempio descrive tutte le connessioni VPC peering.

Comando:

```
aws ec2 describe-vpc-peering-connections
```

Output:

```
{
  "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"
    }
  }
]
}

```

Per descrivere connessioni di VPC peering specifiche

Questo esempio descrive tutte le connessioni VPC peering che si trovano nello stato di accettazione in sospeso.

Comando:

```
aws ec2 describe-vpc-peering-connections --filters Name=status-code,Values=pending-acceptance
```

Questo esempio descrive tutte le connessioni VPC peering con il tag Owner=Finance.

Comando:

```
aws ec2 describe-vpc-peering-connections --filters Name=tag:Owner,Values=Finance
```

Questo esempio descrive tutte le connessioni VPC peering richieste per il vpc-1a2b3c4d specificato. VPC

Comando:

```
aws ec2 describe-vpc-peering-connections --filters Name=requester-vpc-info.vpc-id,Values=vpc-1a2b3c4d
```

- API Per [DescribeVpcPeeringConnections AWS CLI](#) i dettagli, vedere in Command Reference.

## describe-vpcs

Il seguente esempio di codice mostra come utilizzare `describe-vpcs`.

AWS CLI

Esempio 1: Per descrivere tutti i tuoi VPCs

L'`describe-vpcs` esempio seguente recupera i dettagli sul tuo VPCs.

```
aws ec2 describe-vpcs
```

Output:

```
{
  "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"
    }
  ]
}
```

```

    ]
  }
]
}

```

Esempio 2: Per descrivere uno specifico VPC

L' `describe-vpcs` seguente recupera i dettagli per quanto specificato VPC.

```

aws ec2 describe-vpcs \
  --vpc-ids vpc-06e4ab6c6cEXAMPLE

```

Output:

```

{
  "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"
        }
      ]
    }
  ]
}

```

- Per API i dettagli, vedere [DescribeVpcs](#) in AWS CLI Command Reference.

## describe-vpn-connections

Il seguente esempio di codice mostra come utilizzare `describe-vpn-connections`.

### AWS CLI

Esempio 1: Per descrivere le VPN connessioni

L'`describe-vpn-connections` esempio seguente descrive tutte le Site-to-Site VPN connessioni.

```
aws ec2 describe-vpn-connections
```

Output:

```
{
  "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": [
        {
```





Comando:

```
aws ec2 describe-vpn-gateways
```

Output:

```
{
  "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"
        }
      ]
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeVpnGateways](#) in AWS CLI Command Reference.

## detach-classic-link-vpc

Il seguente esempio di codice mostra come utilizzare `detach-classic-link-vpc`.

AWS CLI

Per scollegare (scollegare) un'istanza EC2 -Classic da un VPC

Questo esempio scollega l'istanza `i-0598c7d356eba48d7` da `vpc-88888888`. VPC

Comando:

```
aws ec2 detach-classic-link-vpc --instance-id i-0598c7d356eba48d7 --vpc-id vpc-88888888
```

Output:

```
{  
  "Return": true  
}
```

- Per API i dettagli, [DetachClassicLinkVpc](#) vedere in Command Reference.AWS CLI

## **detach-internet-gateway**

Il seguente esempio di codice mostra come utilizzare `detach-internet-gateway`.

AWS CLI

Per scollegare un gateway Internet dal tuo VPC

L'esempio seguente separa il gateway Internet specificato dallo specifico VPC.

```
aws ec2 detach-internet-gateway \  
  --internet-gateway-id igw-0d0fb496b3EXAMPLE \  
  --vpc-id vpc-0a60eb65b4EXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Internet gateway](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [DetachInternetGateway AWS CLI](#) Command Reference.

## **detach-network-interface**

Il seguente esempio di codice mostra come utilizzare `detach-network-interface`.

## AWS CLI

Per scollegare un'interfaccia di rete dall'istanza

Questo esempio scollega l'interfaccia di rete specificata dall'istanza specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 detach-network-interface --attachment-id eni-attach-66c4350a
```

- Per API i dettagli, vedere [DetachNetworkInterface](#) in AWS CLI Command Reference.

## detach-verified-access-trust-provider

Il seguente esempio di codice mostra come utilizzare `detach-verified-access-trust-provider`.

### AWS CLI

Per scollegare un fornitore di fiducia da un'istanza

L'esempio seguente scollega il provider fiduciario di accesso verificato specificato dall'istanza di accesso verificato specificata.

```
aws ec2 detach-verified-access-trust-provider \  
  --verified-access-instance-id vai-0ce000c0b7643abea \  
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

Output:

```
{  
  "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"
  }
}
```

Per ulteriori informazioni, consulta le [istanze di accesso verificato nella Guida](#) per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [DetachVerifiedAccessTrustProvider AWS CLI Command Reference](#).

## detach-volume

Il seguente esempio di codice mostra come utilizzare `detach-volume`.

### AWS CLI

Per scollegare un volume da un'istanza

Questo comando di esempio scollega il volume (`vol-049df61146c4d7901`) dall'istanza a cui è collegato.

Comando:

```
aws ec2 detach-volume --volume-id vol-1234567890abcdef0
```

Output:

```
{
  "AttachTime": "2014-02-27T19:23:06.000Z",
  "InstanceId": "i-1234567890abcdef0",
  "VolumeId": "vol-049df61146c4d7901",
  "State": "detaching",
  "Device": "/dev/sdb"
}
```

- Per API i dettagli, vedete [DetachVolume](#) in AWS CLI Command Reference.

## detach-vpn-gateway

Il seguente esempio di codice mostra come utilizzare `detach-vpn-gateway`.

### AWS CLI

Per scollegare un gateway privato virtuale dal tuo VPC

Questo esempio scollega il gateway privato virtuale specificato dal gateway specificato. VPC Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 detach-vpn-gateway --vpn-gateway-id vgw-9a4cacf3 --vpc-id vpc-a01106c2
```

- Per API i dettagli, vedere [DetachVpnGateway](#) in AWS CLI Command Reference.

## disable-address-transfer

Il seguente esempio di codice mostra come utilizzare `disable-address-transfer`.

### AWS CLI

Per disabilitare il trasferimento di un indirizzo IP elastico

L'`disable-address-transfer` seguente esempio disabilita il trasferimento elastico dell'indirizzo IP per l'indirizzo IP elastico specificato.

```
aws ec2 disable-address-transfer \  
  --allocation-id eipalloc-09ad461b0d03f6aaf
```

Output:

```
{  
  "AddressTransfer": {  
    "PublicIp": "100.21.184.216",  
    "AllocationId": "eipalloc-09ad461b0d03f6aaf",  
    "AddressTransferStatus": "disabled"  
  }  
}
```

Per ulteriori informazioni, consulta [Transfer Elastic IP address](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [DisableAddressTransfer AWS CLI Command Reference](#).

## **disable-aws-network-performance-metric-subscription**

Il seguente esempio di codice mostra come utilizzare `disable-aws-network-performance-metric-subscription`.

### AWS CLI

Per disabilitare un abbonamento metrico

L'esempio seguente disabilita il monitoraggio della latenza di rete aggregata tra le regioni di origine e di destinazione specificate.

```
aws ec2 disable-aws-network-performance-metric-subscription \
  --source us-east-1 \
  --destination eu-west-1 \
  --metric aggregate-latency \
  --statistic p50
```

Output:

```
{
  "Output": true
}
```

Per ulteriori informazioni, consulta [Manage subscriptions](#) nella Infrastructure Performance User Guide.

- Per API i dettagli, vedere [DisableAwsNetworkPerformanceMetricSubscription](#) in AWS CLI Command Reference.

## **disable-efs-encryption-by-default**

Il seguente esempio di codice mostra come utilizzare `disable-efs-encryption-by-default`.

### AWS CLI

Per disabilitare EFS la crittografia per impostazione predefinita

L'`disable-ebs-encryption-by-default` seguente disabilita EBS la crittografia per impostazione predefinita per il tuo AWS account nella regione corrente.

```
aws ec2 disable-ebs-encryption-by-default
```

Output:

```
{
  "EbsEncryptionByDefault": false
}
```

- Per API i dettagli, vedere [DisableEbsEncryptionByDefault](#) in AWS CLI Command Reference.

## disable-fast-launch

Il seguente esempio di codice mostra come utilizzare `disable-fast-launch`.

AWS CLI

Per interrompere l'avvio rapido di un'immagine

L'`disable-fast-launch` seguente interrompe l'avvio rapido sull'oggetto specificato e pulisce le AMI istantanee preimpostate esistenti.

```
aws ec2 disable-fast-launch \
  --image-id ami-01234567890abcdef
```

Output:

```
{
  "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"
  }

```

Per ulteriori informazioni sulla configurazione di Windows AMI per un avvio più rapido, consulta [Configure your AMI for faster launching](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta Command [DisableFastLaunch](#)Reference AWS CLI .

## disable-fast-snapshot-restores

Il seguente esempio di codice mostra come utilizzare `disable-fast-snapshot-restores`.

### AWS CLI

Per disabilitare il ripristino rapido delle istantanee

L'esempio seguente disabilita il ripristino rapido delle istantanee per l'istananea specificata nella zona di disponibilità specificata.

```

aws ec2 disable-fast-snapshot-restores \
  --availability-zones us-east-2a \
  --source-snapshot-ids snap-1234567890abcdef0

```

Output:

```

{
  "Successful": [
    {
      "SnapshotId": "snap-1234567890abcdef0",
      "AvailabilityZone": "us-east-2a",
      "State": "disabling",
      "StateTransitionReason": "Client.UserInitiated",
      "OwnerId": "123456789012",
      "EnablingTime": "2020-01-25T23:57:49.602Z"
    }
  ],
  "Unsuccessful": []
}

```

- Per API i dettagli, vedere [DisableFastSnapshotRestores](#)in AWS CLI Command Reference.

## disable-image-block-public-access

Il seguente esempio di codice mostra come utilizzare `disable-image-block-public-access`.

### AWS CLI

Per disabilitare l'accesso pubblico a blocchi AMIs nella regione specificata

L'`disable-image-block-public-access` esempio seguente disabilita l'accesso pubblico a blocchi AMIs a livello di account nella regione specificata.

```
aws ec2 disable-image-block-public-access \
  --region us-east-1
```

Output:

```
{
  "ImageBlockPublicAccessState": "unblocked"
}
```

Per ulteriori informazioni, consulta [Bloccare l'accesso pubblico al tuo AMIs](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DisableImageBlockPublicAccess AWS CLI](#) Command Reference.

## disable-image-deprecation

Il seguente esempio di codice mostra come utilizzare `disable-image-deprecation`.

### AWS CLI

Per annullare la deprecazione di un AMI

L'`disable-image-deprecation` esempio seguente annulla la deprecazione di un AMI, che rimuove il campo dall'output. `DeprecationTime` `describe-images` È necessario essere il AMI proprietario per eseguire questa procedura.

```
aws ec2 disable-image-deprecation \
  --image-id ami-1234567890abcdef0
```

Output:

```
{
  "RequestID": "11aabb229-4eac-35bd-99ed-be587EXAMPLE",
  "Return": "true"
}
```

Per ulteriori informazioni, consulta [Deprecare un AMI](https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/ami-deprecate.html#deprecate-ami) <[https://docs.aws.amazon.com/AWS\\_EC2/latest/UserGuide/ami-deprecate.html#deprecate-ami](https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/ami-deprecate.html#deprecate-ami)> nella Amazon User Guide. EC2

- Per i API dettagli, consulta [DisableImageDeprecation](#) Command Reference.AWS CLI

## disable-image

Il seguente esempio di codice mostra come utilizzare `disable-image`.

### AWS CLI

Per disabilitare un AMI

L'`disable-image`esempio seguente disabilita il valore specificatoAMI.

```
aws ec2 disable-image \
  --image-id ami-1234567890abcdef0
```

Output:

```
{
  "Return": "true"
}
```

Per ulteriori informazioni, consulta [Disabilita un AMI](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DisableImage AWS CLI](#) Command Reference.

## disable-ipam-organization-admin-account

Il seguente esempio di codice mostra come utilizzare `disable-ipam-organization-admin-account`.

### AWS CLI

Per disabilitare l'amministratore delegato IPAM

In alcuni scenari, ti integrerai IPAM con AWS Organizations. In tal caso, l'account di gestione AWS Organizations delega un account membro di AWS Organizations come IPAM amministratore.

In questo esempio, sei l'account di gestione AWS Organizations che ha delegato l'account IPAM amministratore e desideri disabilitare tale account dall'essere IPAM amministratore.

Puoi utilizzare qualsiasi AWS regione per `--region` effettuare questa richiesta. Non è necessario utilizzare la regione in cui originariamente hai delegato l'amministratore, in cui IPAM è stato creato o una regione IPAM operativa. Se disabiliti l'account amministratore delegato, puoi riattivarlo in qualsiasi momento o delegare un nuovo account come amministratore. IPAM

L'`disable-ipam-organization-admin-account` seguente disabilita l'amministratore delegato nel tuo account. IPAM AWS

```
aws ec2 disable-ipam-organization-admin-account \
  --delegated-admin-account-id 320805250157 \
  --region ap-south-1
```

Output:

```
{
  "Success": true
}
```

Per ulteriori informazioni, consulta [Integrazione IPAM con gli account di un' AWS organizzazione](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [DisableIpamOrganizationAdminAccount AWS CLI Command Reference](#).

## **disable-serial-console-access**

Il seguente esempio di codice mostra come utilizzare `disable-serial-console-access`.

AWS CLI

Per disabilitare l'accesso alla console EC2 seriale per il tuo account

L'`disable-serial-console-access` seguente disabilita l'accesso tramite account alla console seriale.

```
aws ec2 disable-serial-console-access
```

Output:

```
{
  "SerialConsoleAccessEnabled": false
}
```

Per ulteriori informazioni, consulta [EC2Serial Console](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DisableSerialConsoleAccess AWS CLI](#) Command Reference.

## **disable-snapshot-block-public-access**

Il seguente esempio di codice mostra come utilizzare `disable-snapshot-block-public-access`.

AWS CLI

Per disabilitare l'accesso pubblico a blocchi per le istantanee

L'esempio seguente disabilita l'accesso pubblico a blocchi per le istantanee per consentire la condivisione pubblica delle istantanee.

```
aws ec2 disable-snapshot-block-public-access
```

Output:

```
{
  "State": "unblocked"
}
```

Per ulteriori informazioni, consulta [Bloccare l'accesso pubblico per le istantanee](#) nella Amazon EBS User Guide.

- Per API i dettagli, consulta [DisableSnapshotBlockPublicAccess AWS CLI](#) Command Reference.

## **disable-transit-gateway-route-table-propagation**

Il seguente esempio di codice mostra come utilizzare `disable-transit-gateway-route-table-propagation`.

## AWS CLI

Per disabilitare un allegato del gateway di transito per propagare le rotte alla tabella delle rotte di propagazione specificata

L'esempio seguente disabilita l'allegato specificato per propagare le rotte alla tabella delle rotte di propagazione specificata.

```
aws ec2 disable-transit-gateway-route-table-propagation \  
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \  
  --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

Output:

```
{  
  "Propagation": {  
    "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",  
    "ResourceId": "vpc-4d7de228",  
    "ResourceType": "vpc",  
    "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",  
    "State": "disabled"  
  }  
}
```

Per ulteriori informazioni, vedete le [tabelle di routing dei gateway di transito](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [DisableTransitGatewayRouteTablePropagation](#) in AWS CLI Command Reference.

## disable-vgw-route-propagation

Il seguente esempio di codice mostra come utilizzare `disable-vgw-route-propagation`.

## AWS CLI

Per disabilitare la propagazione delle rotte

Questo esempio disabilita il gateway privato virtuale specificato dalla propagazione delle route statiche alla tabella di route specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 disable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-id vgw-9a4cacf3
```

- Per API i dettagli, vedere [DisableVgwRoutePropagation](#) in AWS CLI Command Reference.

## **disable-vpc-classic-link-dns-support**

Il seguente esempio di codice mostra come utilizzare `disable-vpc-classic-link-dns-support`.

AWS CLI

Per disabilitare ClassicLink DNS il supporto per un VPC

Questo esempio disabilita ClassicLink DNS il supporto per `vpc-88888888`.

Comando:

```
aws ec2 disable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

Output:

```
{  
  "Return": true  
}
```

- Per API i dettagli, vedere [DisableVpcClassicLinkDnsSupport](#) in AWS CLI Command Reference.

## **disable-vpc-classic-link**

Il seguente esempio di codice mostra come utilizzare `disable-vpc-classic-link`.

AWS CLI

Da disabilitare ClassicLink per un VPC

Questo esempio disabilita ClassicLink `vpc-88888888`.

Comando:

```
aws ec2 disable-vpc-classic-link --vpc-id vpc-88888888
```

Output:

```
{  
  "Return": true  
}
```

- Per i API dettagli, vedere [DisableVpcClassicLink](#) in Command Reference.AWS CLI

## disassociate-address

Il seguente esempio di codice mostra come utilizzare `disassociate-address`.

### AWS CLI

Per dissociare un indirizzo IP elastico in EC2 -Classic

Questo esempio dissocia un indirizzo IP elastico da un'istanza in -Classic. EC2 Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 disassociate-address --public-ip 198.51.100.0
```

Per dissociare un indirizzo IP elastico in - EC2 VPC

Questo esempio dissocia un indirizzo IP elastico da un'istanza in a. VPC Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 disassociate-address --association-id eipassoc-2bebb745
```

- Per API i dettagli, vedere [DisassociateAddress](#) in AWS CLI Command Reference.

## disassociate-client-vpn-target-network

Il seguente esempio di codice mostra come utilizzare `disassociate-client-vpn-target-network`.



## AWS CLI

Dissociare una rete da un endpoint Client VPN

L'`disassociate-client-vpn-target-network` seguente dissocia la rete di destinazione associata all'ID di `cvpn-assoc-12312312312312312` associazione per l'endpoint Client specificato. VPN

```
aws ec2 disassociate-client-vpn-target-network \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
  --association-id cvpn-assoc-12312312312312312
```

Output:

```
{
  "AssociationId": "cvpn-assoc-12312312312312312",
  "Status": {
    "Code": "disassociating"
  }
}
```

Per ulteriori informazioni, consulta [Target Networks](#) nella AWS Client VPN Administrator Guide.

- Per API i dettagli, vedere [DisassociateClientVpnTargetNetwork](#) in AWS CLI Command Reference.

## `disassociate-iam-instance-profile`

Il seguente esempio di codice mostra come utilizzare `disassociate-iam-instance-profile`.

### AWS CLI

Per dissociare un profilo di IAM istanza

Questo esempio dissocia un profilo di IAM istanza dall'ID di associazione. `iip-assoc-05020b59952902f5f`

Comando:

```
aws ec2 disassociate-iam-instance-profile --association-id iip-
assoc-05020b59952902f5f
```

**Output:**

```
{
  "IamInstanceProfileAssociation": {
    "InstanceId": "i-123456789abcde123",
    "State": "disassociating",
    "AssociationId": "iip-assoc-05020b59952902f5f",
    "IamInstanceProfile": {
      "Id": "AIPAI5IVIHMFFYY2DKV5Y",
      "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
    }
  }
}
```

- Per API i dettagli, vedere [DisassociateIamInstanceProfile](#) in AWS CLI Command Reference.

**disassociate-instance-event-window**

Il seguente esempio di codice mostra come utilizzare `disassociate-instance-event-window`.

**AWS CLI**

Esempio 1: dissociare una o più istanze da una finestra di evento

L'`disassociate-instance-event-window` esempio seguente dissocia una o più istanze da una finestra degli eventi. Specificate il `instance-event-window-id` parametro per specificare la finestra dell'evento. Per dissociare le istanze, specificate il `association-target` parametro e, per i valori dei parametri, specificate una o più istanze. IDs

```
aws ec2 disassociate-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

**Output:**

```
{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
  }
}
```

```

    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}

```

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella sezione Eventi pianificati della Amazon EC2 User Guide.

Esempio 2: dissociare i tag di istanza dalla finestra di un evento

L'`disassociate-instance-event-window` seguente dissocia i tag di istanza da una finestra di eventi. Specificate il `instance-event-window-id` parametro per specificare la finestra dell'evento. Per dissociare i tag di istanza, specifica il parametro `association-target` e per i valori dei parametri specifica uno o più tag.

```

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}]"

```

Output:

```

{
  "InstanceEventWindow": {
    "InstanceEventWindowId": "iew-0abcdef1234567890",
    "Name": "myEventWindowName",
    "CronExpression": "* 21-23 * * 2,3",
    "AssociationTarget": {
      "InstanceIds": [],
      "Tags": [],
      "DedicatedHostIds": []
    },
    "State": "creating"
  }
}

```

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella sezione Eventi pianificati della Amazon EC2 User Guide.

### Esempio 3: dissociare un host dedicato da una finestra di evento

L'`disassociate-instance-event-window` seguente dissocia un Dedicated Host da una finestra di evento. Specificate il `instance-event-window-id` parametro per specificare la finestra dell'evento. Per dissociare un host dedicato, specificate il `association-target` parametro e, per i valori dei parametri, specificate uno o più host IDs dedicati.

```
aws ec2 disassociate-instance-event-window \  
  --region us-east-1 \  
  --instance-event-window-id iew-0abcdef1234567890 \  
  --association-target DedicatedHostIds=h-029fa35a02b99801d
```

Output:

```
{  
  "InstanceEventWindow": {  
    "InstanceEventWindowId": "iew-0abcdef1234567890",  
    "Name": "myEventWindowName",  
    "CronExpression": "* 21-23 * * 2,3",  
    "AssociationTarget": {  
      "InstanceIds": [],  
      "Tags": [],  
      "DedicatedHostIds": []  
    },  
    "State": "creating"  
  }  
}
```

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella sezione Eventi pianificati della Amazon EC2 User Guide.

- Per API i dettagli, consulta Command [DisassociateInstanceEventWindow](#) Reference AWS CLI .

### **disassociate-ipam-resource-discovery**

Il seguente esempio di codice mostra come utilizzare `disassociate-ipam-resource-discovery`.

AWS CLI

Per dissociare una scoperta di risorse da un IPAM

In questo esempio, sei un account amministratore IPAM delegato e desideri dissociare un IPAM Resource Discovery dal tuo. IPAM Hai eseguito il comando `describe` e hai notato che vuoi dissociarlo dal tuo per fare spazio IPAM ad altre associazioni. `"ResourceDiscoveryStatus": "not-found"`

L'`disassociate-ipam-resource-discovery` seguente dissocia l'individuazione di una IPAM risorsa nel tuo account. AWS

```
aws ec2 disassociate-ipam-resource-discovery \
  --ipam-resource-discovery-association-id ipam-res-disco-assoc-04382a6346357cf82 \
  --region us-east-1
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Integrazione IPAM con account esterni alla tua organizzazione](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [DisassociateIpamResourceDiscovery AWS CLI Command Reference](#).

## **disassociate-nat-gateway-address**

Il seguente esempio di codice mostra come utilizzare `disassociate-nat-gateway-address`.

## AWS CLI

Per dissociare un indirizzo IP elastico da un gateway pubblico NAT

L'esempio seguente dissocia l'indirizzo IP elastico specificato dal gateway pubblico specificato. NAT

```
aws ec2 disassociate-nat-gateway-address \
  --nat-gateway-id nat-1234567890abcdef0 \
  --association-ids eipassoc-0f96bdca17EXAMPLE
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta i [NATgateway](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [DisassociateNatGatewayAddress AWS CLI Command Reference](#).

## disassociate-route-table

Il seguente esempio di codice mostra come utilizzare `disassociate-route-table`.

## AWS CLI

Per dissociare una tabella di rotte

Questo esempio dissocia la tabella di routing specificata dalla sottorete specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 disassociate-route-table --association-id rtbassoc-781d0d1a
```

- Per API i dettagli, vedere [DisassociateRouteTable](#) in AWS CLI Command Reference.

## **disassociate-subnet-cidr-block**

Il seguente esempio di codice mostra come utilizzare `disassociate-subnet-cidr-block`.

AWS CLI

Per dissociare un IPv6 CIDR blocco da una sottorete

Questo esempio dissocia un IPv6 CIDR blocco da una sottorete utilizzando l'ID di associazione del blocco. CIDR

Comando:

```
aws ec2 disassociate-subnet-cidr-block --association-id subnet-cidr-assoc-3aa54053
```

Output:

```
{
  "SubnetId": "subnet-5f46ec3b",
  "Ipv6CidrBlockAssociation": {
    "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
    "AssociationId": "subnet-cidr-assoc-3aa54053",
    "Ipv6CidrBlockState": {
      "State": "disassociating"
    }
  }
}
```

- Per API i dettagli, vedere [DisassociateSubnetCidrBlock](#) in AWS CLI Command Reference.

## **disassociate-transit-gateway-multicast-domain**

Il seguente esempio di codice mostra come utilizzare `disassociate-transit-gateway-multicast-domain`.

## AWS CLI

Per dissociare le sottoreti da un dominio multicast

L'operazione `disassociate-transit-gateway-multicast-domain` seguente dissocia una sottorete dal dominio multicast specificato.

```
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
```

Output:

```
{
  "Associations": [
    {
      "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",
      "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
      "ResourceId": "vpc-7EXAMPLE",
      "ResourceType": "vpc",
      "Subnets": [
        {
          "SubnetId": "subnet-000de86e3bEXAMPLE",
          "State": "disassociating"
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, vedere [Working with multicast](#) nella Transit Gateways Guide '.

- Per API i dettagli, vedere [DisassociateTransitGatewayMulticastDomain](#) in AWS CLI Command Reference.

## **disassociate-transit-gateway-route-table**

Il seguente esempio di codice mostra come utilizzare `disassociate-transit-gateway-route-table`.

## AWS CLI

Per dissociare una tabella di routing del gateway di transito da un allegato di risorse



L'`disassociate-transit-gateway-route-table` seguente dissocia l'allegato specificato dalla tabella delle rotte del gateway di transito.

```
aws ec2 disassociate-transit-gateway-route-table \
  --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \
  --transit-gateway-attachment-id tgw-attach-08e0bc912cEXAMPLE
```

Output:

```
{
  "Association": {
    "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
    "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",
    "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
    "ResourceType": "direct-connect-gateway",
    "State": "disassociating"
  }
}
```

Per ulteriori informazioni, consulta le [tabelle di routing dei gateway di transito](#) nella Guida ai gateway di transito.

- Per API i dettagli, vedere [DisassociateTransitGatewayRouteTable](#) in AWS CLI Command Reference.

## `disassociate-vpc-cidr-block`

Il seguente esempio di codice mostra come utilizzare `disassociate-vpc-cidr-block`.

### AWS CLI

Per dissociare un IPv6 CIDR blocco da un VPC

Questo esempio dissocia un IPv6 CIDR blocco dall'VPC utilizzo dell'ID di associazione per il blocco. CIDR

Comando:

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-eca54085
```

Output:

```
{
  "Ipv6CidrBlockAssociation": {
    "Ipv6CidrBlock": "2001:db8:1234:1a00::/56",
    "AssociationId": "vpc-cidr-assoc-eca54085",
    "Ipv6CidrBlockState": {
      "State": "disassociating"
    }
  },
  "VpcId": "vpc-a034d6c4"
}
```

Per dissociare un blocco da un IPv4 CIDR VPC

Questo esempio dissocia un IPv4 CIDR blocco da un VPC

Comando:

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-0287ac6b
```

Output:

```
{
  "CidrBlockAssociation": {
    "AssociationId": "vpc-cidr-assoc-0287ac6b",
    "CidrBlock": "172.18.0.0/16",
    "CidrBlockState": {
      "State": "disassociating"
    }
  },
  "VpcId": "vpc-27621243"
}
```

- Per API i dettagli, vedere [DisassociateVpcCidrBlock](#) in AWS CLI Command Reference.

## enable-address-transfer

Il seguente esempio di codice mostra come utilizzare `enable-address-transfer`.

AWS CLI

Per abilitare un trasferimento elastico di indirizzi IP

L'enable-address-transfer esempio seguente abilita il trasferimento elastico dell'indirizzo IP elastico dall'indirizzo IP elastico specificato all'account specificato.

```
aws ec2 enable-address-transfer \  
  --allocation-id eipalloc-09ad461b0d03f6aaf \  
  --transfer-account-id 123456789012
```

Output:

```
{  
  "AddressTransfer": {  
    "PublicIp": "100.21.184.216",  
    "AllocationId": "eipalloc-09ad461b0d03f6aaf",  
    "TransferAccountId": "123456789012",  
    "TransferOfferExpirationTimestamp": "2023-02-22T20:51:01.000Z",  
    "AddressTransferStatus": "pending"  
  }  
}
```

Per ulteriori informazioni, consulta [Transfer Elastic IP address](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [EnableAddressTransfer AWS CLI Command Reference](#).

## enable-aws-network-performance-metric-subscription

Il seguente esempio di codice mostra come utilizzare enable-aws-network-performance-metric-subscription.

AWS CLI

Per abilitare un abbonamento metrico

L'enable-aws-network-performance-metric-subscription esempio seguente abilita il monitoraggio della latenza di rete aggregata tra le regioni di origine e di destinazione specificate.

```
aws ec2 enable-aws-network-performance-metric-subscription \  
  --source us-east-1 \  
  --destination eu-west-1 \  
  --metric aggregate-latency \  
  --statistic p50
```

Output:

```
{
  "Output": true
}
```

Per ulteriori informazioni, consulta [Manage subscriptions](#) nella Infrastructure Performance User Guide.

- Per API i dettagli, vedere [EnableAwsNetworkPerformanceMetricSubscription](#) in AWS CLI Command Reference.

### **enable-ebs-encryption-by-default**

Il seguente esempio di codice mostra come utilizzare `enable-ebs-encryption-by-default`.

AWS CLI

Per abilitare EBS la crittografia per impostazione predefinita

L'esempio seguente abilita EBS la crittografia per impostazione predefinita per il tuo AWS account nella regione corrente.

```
aws ec2 enable-ebs-encryption-by-default
```

Output:

```
{
  "EbsEncryptionByDefault": true
}
```

- Per API i dettagli, vedere [EnableEbsEncryptionByDefault](#) in AWS CLI Command Reference.

### **enable-fast-launch**

Il seguente esempio di codice mostra come utilizzare `enable-fast-launch`.

AWS CLI

Per iniziare l'avvio rapido di un'immagine

L'enable-fast-launchesempio seguente avvia l'avvio rapido sull'area specificata AMI e imposta il numero massimo di istanze parallele da avviare su 6. Il tipo di risorsa da utilizzare per il provisioning AMI è impostato susnapshot, che è anche il valore predefinito.

```
aws ec2 enable-fast-launch \
  --image-id ami-01234567890abcdef \
  --max-parallel-launches 6 \
  --resource-type snapshot
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni sulla configurazione di Windows AMI per un avvio più rapido, consulta [Configure your AMI for faster launching](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta Command [EnableFastLaunch](#)Reference AWS CLI .

## enable-fast-snapshot-restores

Il seguente esempio di codice mostra come utilizzareenable-fast-snapshot-restores.

AWS CLI

Per abilitare il ripristino rapido delle istantanee

L'enable-fast-snapshot-restoresesempio seguente abilita il ripristino rapido delle istantanee per l'istananea specificata nelle zone di disponibilità specificate.

```
aws ec2 enable-fast-snapshot-restores \
```

```
--availability-zones us-east-2a us-east-2b \  
--source-snapshot-ids snap-1234567890abcdef0
```

Output:

```
{  
  "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": []  
}
```

- Per API i dettagli, vedere [EnableFastSnapshotRestores](#) in AWS CLI Command Reference.

## enable-image-block-public-access

Il seguente esempio di codice mostra come utilizzare `enable-image-block-public-access`.

AWS CLI

Per abilitare il blocco dell'accesso pubblico AMIs nella regione specificata

L'esempio seguente abilita il blocco dell'accesso pubblico AMIs a livello di account nella regione specificata.

```
aws ec2 enable-image-block-public-access \  

```

```
--region us-east-1 \  
--image-block-public-access-state block-new-sharing
```

Output:

```
{  
  "ImageBlockPublicAccessState": "block-new-sharing"  
}
```

Per ulteriori informazioni, consulta [Bloccare l'accesso pubblico al tuo AMIs](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [EnableImageBlockPublicAccess AWS CLI Command Reference](#).

## enable-image-deprecation

Il seguente esempio di codice mostra come utilizzare `enable-image-deprecation`.

AWS CLI

Esempio 1: deprecare un AMI

L'esempio seguente rende obsoleto un AMI in una data e un'ora specifiche. Se specifichi un valore in secondi, Amazon EC2 arrotonda i secondi al minuto più vicino. Devi essere il proprietario dell'AMI per eseguire questa procedura.

```
aws ec2 enable-image-deprecation \  
  --image-id ami-1234567890abcdef0 \  
  --deprecate-at "2022-10-15T13:17:12.000Z"
```

Output:

```
{  
  "RequestID": "59dbff89-35bd-4eac-99ed-be587EXAMPLE",  
  "Return": "true"  
}
```

Per ulteriori informazioni, consulta [Deprecare un AMI <https://docs.aws.amazon.com/AWS\\_EC2/latest/UserGuide/ami-deprecate.html#deprecate-ami>](https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/ami-deprecate.html#deprecate-ami) nella Amazon User Guide. EC2

- Per i API dettagli, consulta [EnableImageDeprecation](#) Command Reference.AWS CLI

## **enable-image**

Il seguente esempio di codice mostra come utilizzare `enable-image`.

### AWS CLI

Per abilitare un AMI

L'esempio seguente abilita lo specificato AMI.

```
aws ec2 enable-image \  
  --image-id ami-1234567890abcdef0
```

Output:

```
{  
  "Return": "true"  
}
```

Per ulteriori informazioni, consulta [Disabilita un AMI](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [EnableImage AWS CLI](#) Command Reference.

## **enable-ipam-organization-admin-account**

Il seguente esempio di codice mostra come utilizzare `enable-ipam-organization-admin-account`.

### AWS CLI

Effettuare l'integrazione con AWS Organizations e delegare un account membro come account IPAM

L'esempio seguente si integra IPAM con AWS Organizations e delega un account membro come account. IPAM

```
aws ec2 enable-ipam-organization-admin-account \  
  --delegated-admin-account-id 320805250157
```



Output:

```
{
  "Success": true
}
```

Per ulteriori informazioni, consulta [Integrate IPAM with AWS Organizations](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [EnableIpamOrganizationAdminAccount AWS CLI](#) Command Reference.

### **enable-reachability-analyzer-organization-sharing**

Il seguente esempio di codice mostra come utilizzare `enable-reachability-analyzer-organization-sharing`.

AWS CLI

Per abilitare l'accesso affidabile per Reachability Analyzer

L'esempio seguente abilita l'accesso affidabile per Reachability Analyzer.

```
aws ec2 enable-reachability-analyzer-organization-sharing
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Analisi tra account nella Reachability Analyzer User Guide](#).

- Per i API dettagli, vedere [EnableReachabilityAnalyzerOrganizationSharing](#) in Command Reference.AWS CLI

### **enable-serial-console-access**

Il seguente esempio di codice mostra come utilizzare `enable-serial-console-access`.

AWS CLI

Per abilitare l'accesso alla console seriale per il tuo account

L'enable-serial-console-accessesempio seguente abilita l'accesso tramite account alla console seriale.

```
aws ec2 enable-serial-console-access
```

Output:

```
{
  "SerialConsoleAccessEnabled": true
}
```

Per ulteriori informazioni, consulta [EC2Serial Console](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [EnableSerialConsoleAccess AWS CLI](#) Command Reference.

## enable-snapshot-block-public-access

Il seguente esempio di codice mostra come utilizzareenable-snapshot-block-public-access.

AWS CLI

Per abilitare l'accesso pubblico a blocchi per le istantanee

L'enable-snapshot-block-public-accessesempio seguente blocca tutte le condivisioni pubbliche delle istantanee.

```
aws ec2 enable-snapshot-block-public-access \
  --state block-all-sharing
```

Output:

```
{
  "State": "block-all-sharing"
}
```

Per ulteriori informazioni, consulta [Bloccare l'accesso pubblico per le istantanee](#) nella Amazon EBS User Guide.

- Per API i dettagli, consulta [EnableSnapshotBlockPublicAccess AWS CLI](#) Command Reference.

## enable-transit-gateway-route-table-propagation

Il seguente esempio di codice mostra come utilizzare `enable-transit-gateway-route-table-propagation`.

### AWS CLI

Per consentire a un allegato del gateway di transito di propagare le rotte alla tabella delle rotte di propagazione specificata

L'esempio seguente consente all'allegato specificato di propagare le rotte alla tabella delle rotte di propagazione specificata.

```
aws ec2 enable-transit-gateway-route-table-propagation \
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
  --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

Output:

```
{
  "Propagation": {
    "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
    "ResourceId": "vpc-4d7de228",
    "ResourceType": "vpc",
    "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",
    "State": "disabled"
  }
}
```

Per ulteriori informazioni, vedete le [tabelle di routing dei gateway di transito](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [EnableTransitGatewayRouteTablePropagation](#) in AWS CLI Command Reference.

## enable-vgw-route-propagation

Il seguente esempio di codice mostra come utilizzare `enable-vgw-route-propagation`.

### AWS CLI

Per abilitare la propagazione delle rotte

Questo esempio consente al gateway privato virtuale specificato di propagare le route statiche alla tabella di route specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 enable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-id vgw-9a4cacf3
```

- Per API i dettagli, vedere [EnableVgwRoutePropagation](#) in AWS CLI Command Reference.

## **enable-volume-io**

Il seguente esempio di codice mostra come utilizzare `enable-volume-io`.

AWS CLI

Per abilitare l'I/O per un volume

Questo esempio abilita l'I/O su volume. `vol-1234567890abcdef0`

Comando:

```
aws ec2 enable-volume-io --volume-id vol-1234567890abcdef0
```

Output:

```
{  
  "Return": true  
}
```

- Per API i dettagli, vedere [EnableVolumeIo](#) in AWS CLI Command Reference.

## **enable-vpc-classic-link-dns-support**

Il seguente esempio di codice mostra come utilizzare `enable-vpc-classic-link-dns-support`.

AWS CLI

Per abilitare ClassicLink DNS il supporto per un VPC

Questo esempio abilita ClassicLink DNS il supporto pervpc-88888888.

Comando:

```
aws ec2 enable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

Output:

```
{
  "Return": true
}
```

- Per API i dettagli, vedere [EnableVpcClassicLinkDnsSupport](#) in AWS CLI Command Reference.

## **enable-vpc-classic-link**

Il seguente esempio di codice mostra come utilizzare `enable-vpc-classic-link`.

AWS CLI

Per abilitare un VPC for ClassicLink

Questo esempio abilita vpc-88888888 per. ClassicLink

Comando:

```
aws ec2 enable-vpc-classic-link --vpc-id vpc-88888888
```

Output:

```
{
  "Return": true
}
```

- Per API i dettagli, vedere [EnableVpcClassicLink](#) in AWS CLI Command Reference.

## **export-client-vpn-client-certificate-revocation-list**

Il seguente esempio di codice mostra come utilizzare `export-client-vpn-client-certificate-revocation-list`.

## AWS CLI

Per esportare un elenco di revoche di certificati client

L'esempio seguente esporta l'elenco delle revoche dei certificati client per l'endpoint Client VPN specificato. In questo esempio, l'output viene restituito in formato testo per facilitarne la lettura.

```
aws ec2 export-client-vpn-client-certificate-revocation-list \
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
  --output text
```

Output:

```
-----BEGIN X509 CRL-----
MIICiTCcAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMakGA1UEBhMC
VVMxCzAJBgNVBAsTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBA5TC0lBTsBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHmAd
BgkqhkiG9w0BCQEWEG5vb251QGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI1MjA0NTIxWjCBiDELMakGA1UEBhMCVVMxCzAJBgNVBAsTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC0lBTsBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMxHmAdBgkqhkiG9w0BCQEWEG5vb251QGFT
YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTC2XADZ4nB+BLygVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUHVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END X509 CRL-----
STATUS      pending
```

Per ulteriori informazioni, vedere [Elenchi di revoca dei certificati client](#) nella AWS Client VPN Administrator Guide.

- Per API i dettagli, vedere [ExportClientVpnClientCertificateRevocationList](#) in AWS CLI Command Reference.

### export-client-vpn-client-configuration

Il seguente esempio di codice mostra come utilizzare `export-client-vpn-client-configuration`.

## AWS CLI

Per esportare la configurazione del client

L'export-client-vpn-client-configurationesempio seguente esporta la configurazione del client per l'VPNendpoint Client specificato. In questo esempio, l'output viene restituito in formato testo per facilitarne la lettura.

```
aws ec2 export-client-vpn-client-configuration \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --output text
```

Output:

```
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-----  
MIICiTCCAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC  
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6  
b24xFDASBgNVBAwTC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAd  
BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN  
MTIwNDI1MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD  
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC0lBTSBDb25z  
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAdBgkqhkiG9w0BCQEWEG5vb251QGft  
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ  
21uUSfwfEvySWtC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T  
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE  
Ibb30hjZnczvQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4  
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb  
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvjx79LjStB  
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=  
-----END CERTIFICATE-----
```

```
</ca>  
reneg-sec 0
```

Per ulteriori informazioni, consulta [Client VPN Endpoints](#) nella AWS Client VPN Administrator Guide.

- Per API i dettagli, vedere [ExportClientVpnClientConfiguration](#) in AWS CLI Command Reference.

## export-image

Il seguente esempio di codice mostra come utilizzare `export-image`.

### AWS CLI

Per esportare una macchina virtuale da un AMI

L'esempio seguente esporta il contenuto specificato AMI nel bucket specificato nel formato specificato.

```
aws ec2 export-image \  
  --image-id ami-1234567890abcdef0 \  
  --disk-image-format VMDK \  
  --s3-export-location S3Bucket=my-export-bucket,S3Prefix=exports/
```

Output:

```
{  
  "DiskImageFormat": "vmdk",  
  "ExportImageTaskId": "export-ami-1234567890abcdef0"  
  "ImageId": "ami-1234567890abcdef0",  
  "RoleName": "vmimport",  
  "Progress": "0",  
  "S3ExportLocation": {  
    "S3Bucket": "my-export-bucket",  
    "S3Prefix": "exports/"  
  },  
  "Status": "active",  
  "StatusMessage": "validating"  
}
```

- Per API i dettagli, vedere [ExportImage](#) in AWS CLI Command Reference.



## get-associated-ipv6-pool-cidrs

Il seguente esempio di codice mostra come utilizzare `get-associated-ipv6-pool-cidrs`.

### AWS CLI

Per ottenere le associazioni per un pool di IPv6 indirizzi

L'`get-associated-ipv6-pool-cidrs`esempio seguente ottiene le associazioni per il pool di IPv6 indirizzi specificato.

```
aws ec2 get-associated-ipv6-pool-cidrs \  
  --pool-id ipv6pool-ec2-012345abc12345abc
```

Output:

```
{  
  "Ipv6CidrAssociations": [  
    {  
      "Ipv6Cidr": "2001:db8:1234:1a00::/56",  
      "AssociatedResource": "vpc-111111222222333ab"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [GetAssociatedIpv6 PoolCidrs](#) in AWS CLI Command Reference.

## get-aws-network-performance-data

Il seguente esempio di codice mostra come utilizzare `get-aws-network-performance-data`.

### AWS CLI

Per ottenere dati sulle prestazioni della rete

L'`get-aws-network-performance-data`esempio seguente recupera i dati sulle prestazioni di rete tra le regioni specificate nel periodo di tempo specificato.

```
aws ec2 get-aws-network-performance-data \  
  --start-time 2022-10-26T12:00:00.000Z \  
  --end-time 2022-10-26T12:30:00.000Z \  
  --regions us-east-1 us-west-2
```

```
--data-queries Id=my-query,Source=us-east-1,Destination=eu-west-1,Metric=aggregate-latency,Statistic=p50,Period=five-minutes
```

## Output:

```
{
  "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"
      }
    ]
  }
]
}

```

Per ulteriori informazioni, vedere [Monitor network performance](#) nella Infrastructure Performance User Guide.

- Per API i dettagli, vedere [GetAwsNetworkPerformanceData](#) in AWS CLI Command Reference.

## get-capacity-reservation-usage

Il seguente esempio di codice mostra come utilizzare `get-capacity-reservation-usage`.

### AWS CLI

Per visualizzare l'utilizzo della prenotazione della capacità tra AWS gli account

L'`get-capacity-reservation-usage` esempio seguente visualizza le informazioni sull'utilizzo per la prenotazione di capacità specificata.

```
aws ec2 get-capacity-reservation-usage \
  --capacity-reservation-id cr-1234abcd56EXAMPLE
```

Output:

```
{
  "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
  "InstanceUsages": [
```

```

    {
      "UsedInstanceCount": 1,
      "AccountId": "123456789012"
    }
  ],
  "AvailableInstanceCount": 4,
  "TotalInstanceCount": 5,
  "State": "active",
  "InstanceType": "t2.medium"
}

```

Per ulteriori informazioni, consulta [Visualizzazione dell'utilizzo della prenotazione di capacità condivisa](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per API i dettagli, consulta AWS CLI Command [GetCapacityReservationUsageReference](#).

## get-coip-pool-usage

Il seguente esempio di codice mostra come utilizzare `get-coip-pool-usage`.

### AWS CLI

Per ottenere l'utilizzo del pool di indirizzi IP di proprietà del cliente

L'`get-coip-pool-usage` seguente ottiene i dettagli di utilizzo per il pool di indirizzi IP di proprietà del cliente specificato.

```

aws ec2 get-coip-pool-usage \
  --pool-id ipv4pool-coip-123a45678bEXAMPLE

```

Output:

```

{
  "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"
}
```

Per ulteriori informazioni, consulta [Indirizzi IP di proprietà del cliente](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, vedere [GetCoipPoolUsage](#) in AWS CLI Command Reference.

## get-console-output

Il seguente esempio di codice mostra come utilizzare `get-console-output`.

### AWS CLI

Esempio 1: Per ottenere l'output della console

L'`get-console-output` seguente ottiene l'output della console per l'istanza Linux specificata.

```
aws ec2 get-console-output \
  --instance-id i-1234567890abcdef0
```

Output:

```
{
  "InstanceId": "i-1234567890abcdef0",
  "Timestamp": "2013-07-25T21:23:53.000Z",
  "Output": "...
}
```

Per ulteriori informazioni, consulta l'[output della console di istanza](#) nella Amazon EC2 User Guide.

Esempio 2: per ottenere l'ultimo output della console

L'`get-console-output` seguente ottiene l'ultimo output della console per l'istanza Linux specificata.

```
aws ec2 get-console-output \  
  --instance-id i-1234567890abcdef0 \  
  --latest \  
  --output text
```

Output:

```
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
```

Per ulteriori informazioni, consulta l'[output della console di istanza](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [GetConsoleOutput AWS CLI Command Reference](#).

## get-console-screenshot

Il seguente esempio di codice mostra come utilizzare `get-console-screenshot`.

AWS CLI

Per recuperare uno screenshot di un'istanza in esecuzione

L'`get-console-screenshot` esempio seguente recupera uno screenshot dell'istanza specificata in formato `.jpg`. Lo screenshot viene restituito come stringa con codifica Base64.

```
aws ec2 get-console-screenshot \  
  --instance-id i-1234567890abcdef0
```

Output:

```
{  
  "ImageData": "997987/8kgj49ikjhewkww0008084EXAMPLE",  
  "InstanceId": "i-1234567890abcdef0"
```

```
}
```

- Per i API dettagli, vedere in Command Reference. [GetConsoleScreenshot](#)AWS CLI

## get-default-credit-specification

Il seguente esempio di codice mostra come utilizzare `get-default-credit-specification`.

### AWS CLI

Per descrivere l'opzione di credito predefinita

L'`get-default-credit-specification`esempio seguente descrive l'opzione di credito predefinita per le istanze T2.

```
aws ec2 get-default-credit-specification \  
  --instance-family t2
```

Output:

```
{  
  "InstanceFamilyCreditSpecification": {  
    "InstanceFamily": "t2",  
    "CpuCredits": "standard"  
  }  
}
```

- Per API i dettagli, vedere [GetDefaultCreditSpecification](#)in AWS CLI Command Reference.

## get-ebs-default-kms-key-id

Il seguente esempio di codice mostra come utilizzare `get-ebs-default-kms-key-id`.

### AWS CLI

Per descrivere l'impostazione predefinita CMK per la EBS crittografia

L'`get-ebs-default-kms-key-id`esempio seguente descrive l'impostazione predefinita CMK per la EBS crittografia del tuo AWS account.

```
aws ec2 get-ebs-default-kms-key-id
```

L'output mostra l'impostazione predefinita CMK per la EBS crittografia, che è AWS gestita CMK con l'alias `alias/aws/ebs`.

```
{
  "KmsKeyId": "alias/aws/ebs"
}
```

L'output seguente mostra una EBS crittografia personalizzata CMK.

```
{
  "KmsKeyId": "arn:aws:kms:us-
west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"
}
```

- Per API i dettagli, vedere [GetEbsDefaultKmsKeyId](#) in AWS CLI Command Reference.

## get-ebs-encryption-by-default

Il seguente esempio di codice mostra come utilizzare `get-ebs-encryption-by-default`.

### AWS CLI

Per descrivere se EBS la crittografia è abilitata per impostazione predefinita

L'`get-ebs-encryption-by-default` esempio seguente indica se EBS la crittografia per impostazione predefinita è abilitata per l' AWS account nella regione corrente.

```
aws ec2 get-ebs-encryption-by-default
```

L'output seguente indica che la EBS crittografia per impostazione predefinita è disabilitata.

```
{
  "EbsEncryptionByDefault": false
}
```

L'output seguente indica che la EBS crittografia è abilitata per impostazione predefinita.

```
{
  "EbsEncryptionByDefault": true
}
```



```
}

```

- Per API i dettagli, vedere [GetEbsEncryptionByDefault](#) in AWS CLI Command Reference.

## get-flow-logs-integration-template

Il seguente esempio di codice mostra come utilizzare `get-flow-logs-integration-template`.

### AWS CLI

Per creare un CloudFormation modello per automatizzare l'integrazione dei log di VPC flusso con Amazon Athena

I seguenti `get-flow-logs-integration-template` esempi creano un CloudFormation modello per automatizzare l'integrazione dei log di VPC flusso con Amazon Athena.

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
```

Output:

```
{

```

```
"Result": "https://DOC-EXAMPLE-BUCKET.s3.us-east-2.amazonaws.com/
VPCFlowLogsIntegrationTemplate_fl-1234567890abcdef0_Wed%20Jul
%2021%2000%3A57%3A56%20UTC%202021.yml"
}
```

Per informazioni sull'uso dei CloudFormation modelli, consulta [Lavorare con i AWS CloudFormation modelli nella Guida per l'AWS CloudFormation utente](#).

Per informazioni sull'uso di Amazon Athena e dei log di flusso, consulta la sezione [Query flow logs using Amazon Athena nella Amazon Virtual Private Cloud User Guide](#).

- Per i API dettagli, consulta [GetFlowLogsIntegrationTemplate](#) Command Reference.AWS CLI

## get-groups-for-capacity-reservation

Il seguente esempio di codice mostra come utilizzare `get-groups-for-capacity-reservation`.

### AWS CLI

Per elencare i gruppi di risorse con una riserva di capacità

L'`get-groups-for-capacity-reservation` esempio seguente elenca i gruppi di risorse a cui è stata aggiunta la prenotazione di capacità specificata.

```
aws ec2 get-groups-for-capacity-reservation \
  --capacity-reservation-id cr-1234abcd56EXAMPLE
```

Output:

```
{
  "CapacityReservationsGroup": [
    {
      "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/my-
resource-group",
      "OwnerId": "123456789012"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Working with Capacity Reservations](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

- Per API i dettagli, consulta AWS CLI Command [GetGroupsForCapacityReservation](#) Reference.

## get-host-reservation-purchase-preview

Il seguente esempio di codice mostra come utilizzare `get-host-reservation-purchase-preview`.

### AWS CLI

Per visualizzare un'anteprima di acquisto per una prenotazione dedicata agli host

Questo esempio fornisce un'anteprima dei costi di una specifica prenotazione di host dedicato per l'host dedicato specificato nel tuo account.

Comando:

```
aws ec2 get-host-reservation-purchase-preview --offering-id hro-03f707bf363b6b324 --  
host-id-set h-013abcd2a00cbd123
```

Output:

```
{  
  "TotalHourlyPrice": "1.499",  
  "Purchase": [  
    {  
      "HourlyPrice": "1.499",  
      "InstanceFamily": "m4",  
      "PaymentOption": "NoUpfront",  
      "HostIdSet": [  
        "h-013abcd2a00cbd123"  
      ],  
      "UpfrontPrice": "0.000",  
      "Duration": 31536000  
    }  
  ],  
  "TotalUpfrontPrice": "0.000"  
}
```

- Per API i dettagli, vedere [GetHostReservationPurchasePreview](#) in AWS CLI Command Reference.

## get-image-block-public-access-state

Il seguente esempio di codice mostra come utilizzare `get-image-block-public-access-state`.

## AWS CLI

Per ottenere lo stato di blocco dell'accesso pubblico AMIs nella regione specificata

L'`get-image-block-public-access-state` seguente ottiene lo stato di blocco dell'accesso pubblico AMIs a livello di account nella regione specificata.

```
aws ec2 get-image-block-public-access-state \  
  --region us-east-1
```

Output:

```
{  
  "ImageBlockPublicAccessState": "block-new-sharing"  
}
```

Per ulteriori informazioni, consulta [Bloccare l'accesso pubblico al tuo AMIs](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [GetImageBlockPublicAccessState AWS CLI Command Reference](#).

## **get-instance-types-from-instance-requirements**

Il seguente esempio di codice mostra come utilizzare `get-instance-types-from-instance-requirements`.

## AWS CLI

Per visualizzare in anteprima i tipi di istanza che corrispondono agli attributi specificati

L'`get-instance-types-from-instance-requirements` seguente genera innanzitutto un elenco di tutti i possibili attributi che possono essere specificati utilizzando il `--generate-cli-skeleton` parametro e salva l'elenco in un JSON file. Quindi, il JSON file viene utilizzato per personalizzare gli attributi per i quali visualizzare in anteprima i tipi di istanza corrispondenti.

Per generare tutti gli attributi possibili e salvare l'output direttamente in un JSON file, utilizzate il comando seguente.

```
aws ec2 get-instance-types-from-instance-requirements \  
  --generate-cli-skeleton
```

```
--region us-east-1 \  
--generate-cli-skeleton input > attributes.json
```

Output:

```
{  
  "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": ""
}
```

Configura il file JSON. È necessario fornire valori per `ArchitectureTypes`, `VirtualizationTypes`, `VCpuCount` e `MemoryMiB`. È possibile omettere gli altri attributi. Se omissi, vengono utilizzati i valori predefiniti. Per una descrizione di ogni attributo e dei relativi valori predefiniti, vedere `get-instance-types-from -instance-requirements < - -requirements.html>`. <https://docs.aws.amazon.com/cli/latest/reference/ec2/get-instance-types-from-instance>

Visualizza l'anteprima dei tipi di istanza con gli attributi specificati in `attributes.json`. Specificate il nome e il percorso JSON del file utilizzando il `--cli-input-json` parametro. Nella richiesta seguente, l'output viene formattato come tabella.

```
aws ec2 get-instance-types-from-instance-requirements \
  --cli-input-json file://attributes.json \
  --output table
```

Contenuto del `attributes.json` file:

```
{
  "ArchitectureTypes": [
    "x86_64"
  ],
  "VirtualizationTypes": [
    "hvm"
  ],
  "InstanceRequirements": {
    "VCpuCount": {
      "Min": 4,
      "Max": 6
    },
    "MemoryMiB": {
      "Min": 2048
    },
    "InstanceGenerations": [
      "current"
    ]
  }
}
```

Output:

```
-----
|GetInstanceTypesFromInstanceRequirements|
+-----+
||           InstanceTypes           ||
|+-----+|
||           InstanceType           ||
|+-----+|
||  c4.xlarge                        ||
```

```

|| c5.xlarge           ||
|| c5a.xlarge         ||
|| c5ad.xlarge        ||
|| c5d.xlarge         ||
|| c5n.xlarge         ||
|| d2.xlarge          ||
...

```

Per ulteriori informazioni sulla selezione del tipo di istanza basata sugli attributi, consulta [Come funziona la selezione del tipo di istanza basata sugli attributi nella Amazon User Guide](#). EC2

- Per i API dettagli, consulta Command Reference. [GetInstanceTypesFromInstanceRequirements](#) AWS CLI

## get-instance-uefi-data

Il seguente esempio di codice mostra come utilizzare `get-instance-uefi-data`.

### AWS CLI

Per recuperare UEFI dati da un'istanza

L'`get-instance-uefi-data` esempio seguente recupera UEFI i dati da un'istanza. Se l'output è vuoto, l'istanza non contiene UEFI dati.

```

aws ec2 get-instance-uefi-data \
  --instance-id i-0123456789example

```

Output:

```

{
  "InstanceId": "i-0123456789example",
  "UefiData": "QU1aTlVFRkkf+uLXAAAAAHj5a7fZ9+3dBzxXb/.
<snipped>
AAAD4L/J/A0Dshho="
}

```

Per ulteriori informazioni, consulta [UEFI Secure Boot](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [GetInstanceUefiData AWS CLI](#) Command Reference.



## get-ipam-address-history

Il seguente esempio di codice mostra come utilizzare `get-ipam-address-history`.

### AWS CLI

Per ottenere la cronologia di un CIDR

L'`get-ipam-address-history` seguente ottiene la cronologia di un 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
```

Output:

```
{  
  "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"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Visualizza la cronologia degli indirizzi IP](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [GetIpamAddressHistory AWS CLI Command Reference](#).

## get-ipam-discovered-accounts

Il seguente esempio di codice mostra come utilizzare `get-ipam-discovered-accounts`.

### AWS CLI

Per visualizzare gli account scoperti da un IPAM

In questo scenario, sei un amministratore IPAM delegato che desidera visualizzare gli AWS account proprietari delle risorse che IPAM sta scoprendo.

`--discovery-region` Questa è la regione IPAM operativa in cui desideri visualizzare gli stati degli account monitorati. Ad esempio, se hai tre regioni IPAM operative, potresti voler effettuare questa richiesta tre volte per visualizzare i timestamp specifici del rilevamento in ciascuna di quelle aree particolari.

L'`get-ipam-discovered-accounts` esempio seguente elenca gli AWS account che possiedono le risorse che IPAM sta scoprendo.

```
aws ec2 get-ipam-discovered-accounts \  
  --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \  
  --discovery-region us-east-1
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Integrazione IPAM con account esterni alla tua organizzazione](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [GetIpamDiscoveredAccounts AWS CLI Command Reference](#).

## `get-ipam-discovered-public-addresses`

Il seguente esempio di codice mostra come utilizzare `get-ipam-discovered-public-addresses`.

AWS CLI

Per visualizzare gli indirizzi IP pubblici rilevati

In questo esempio, sei un amministratore IPAM delegato e desideri visualizzare gli indirizzi IP delle risorse scoperte dalPAM. Puoi ottenere l'ID di scoperta delle risorse con [describe-ipam-resource-discoveries](#).

L'get-ipam-discovered-public-addresses esempio seguente mostra gli indirizzi IP pubblici rilevati per l'individuazione di una risorsa.

```
aws ec2 get-ipam-discovered-public-addresses \
  --ipam-resource-discovery-id ipam-res-disco-0f4ef577a9f37a162 \
  --address-region us-east-1 \
  --region us-east-1
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Visualizza gli approfondimenti sugli IP pubblici](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [GetIpamDiscoveredPublicAddresses AWS CLI](#) Command Reference.

## get-ipam-discovered-resource-cidrs

Il seguente esempio di codice mostra come utilizzare `get-ipam-discovered-resource-cidrs`.

### AWS CLI

Per visualizzare l'indirizzo IP CIDRs scoperto da un IPAM

In questo esempio, sei un amministratore IPAM delegato che desidera visualizzare i dettagli relativi all'indirizzo CIDRs IP delle risorse che IPAM sta scoprendo.

Per completare questa richiesta:

L'individuazione della risorsa scelta deve essere associata a IPAM. The `--resource-region` è la AWS regione in cui è stata creata la risorsa.

L'`get-ipam-discovered-resource-cidrs` esempio seguente elenca gli indirizzi IP delle risorse che IPAM sta scoprendo.

```
aws ec2 get-ipam-discovered-resource-cidrs \  
  --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \  
  --resource-region us-east-1
```

Output:

```
{  
  {  
    "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"
  }
}
}

```

Per ulteriori informazioni, consulta [Monitora CIDR l'utilizzo per risorsa](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [GetIpamDiscoveredResourceCidrs AWS CLI Command Reference](#).

## get-ipam-pool-allocations

Il seguente esempio di codice mostra come utilizzare `get-ipam-pool-allocations`.

### AWS CLI

Per ottenere l'ID della CIDR da un pool IPAM

L'esempio seguente ottiene l'ID della CIDR da un pool IPAM.

(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

```

Output:

```

{
  "IpamPoolAllocations": [
    {
      "Cidr": "10.0.0.0/16",

```

```

        "IpamPoolAllocationId": "ipam-pool-
alloc-0e6186d73999e47389266a5d6991e6220",
        "ResourceType": "custom",
        "ResourceOwner": "123456789012"
    }
]
}

```

- Per API i dettagli, vedere [GetIpamPoolAllocations](#) in AWS CLI Command Reference.

## get-ipam-pool-cidrs

Il seguente esempio di codice mostra come utilizzare `get-ipam-pool-cidrs`.

### AWS CLI

Per far sì che i CIDRs provisioning venga inserito in un pool IPAM

L'`get-ipam-pool-cidrs` esempio seguente trasferisce i CIDRs provisioning a un IPAM pool.

(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.*

```

Output:

```

{
  "IpamPoolCidr": {
    "Cidr": "10.0.0.0/24",
    "State": "provisioned"
  }
}

```

- Per API i dettagli, vedere [GetIpamPoolCidrs](#) in AWS CLI Command Reference.



## get-ipam-resource-cidrs

Il seguente esempio di codice mostra come utilizzare `get-ipam-resource-cidrs`.

### AWS CLI

Per ottenere l'IDR allocazione a una risorsa

L'`get-ipam-resource-cidrs` esempio seguente ottiene l'IDR allocazione a una risorsa.

(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
```

Output:

```
{  
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Monitora CIDR l'utilizzo per risorsa](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [GetIppamResourceCidrs AWS CLI Command Reference](#).

## get-launch-template-data

Il seguente esempio di codice mostra come utilizzare `get-launch-template-data`.

### AWS CLI

Per ottenere i dati delle istanze per un modello di lancio

Questo esempio ottiene dati sull'istanza specificata e utilizza l'`--query` opzione per restituire il contenuto in `LaunchTemplateData`. È possibile utilizzare l'output come base per creare un nuovo modello di avvio o una nuova versione del modello di avvio.

Comando:

```
aws ec2 get-launch-template-data --instance-id i-0123d646e8048babc --query
'LaunchTemplateData'
```

Output:

```
{
  "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"
        }
    ]
}
```

- Per API i dettagli, vedere [GetLaunchTemplateData](#) in AWS CLI Command Reference.

## get-managed-prefix-list-associations

Il seguente esempio di codice mostra come utilizzare `get-managed-prefix-list-associations`.

## AWS CLI

Per ottenere le associazioni degli elenchi di prefissi

L'get-managed-prefix-list-associationsesempio seguente ottiene le risorse associate all'elenco di prefissi specificato.

```
aws ec2 get-managed-prefix-list-associations \  
  --prefix-list-id pl-0123456abcabcabc1
```

Output:

```
{  
  "PrefixListAssociations": [  
    {  
      "ResourceId": "sg-0abc123456abc12345",  
      "ResourceOwner": "123456789012"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Managed Prefix Lists](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [GetManagedPrefixListAssociations AWS CLI](#) Command Reference.

## get-managed-prefix-list-entries

Il seguente esempio di codice mostra come utilizzareget-managed-prefix-list-entries.

## AWS CLI

Per ottenere le voci per un elenco di prefissi

Quanto segue get-managed-prefix-list-entries ottiene le voci per l'elenco di prefissi specificato.

```
aws ec2 get-managed-prefix-list-entries \  
  --prefix-list-id pl-0123456abcabcabc1
```

Output:

```
{
```

```
"Entries": [  
  {  
    "Cidr": "10.0.0.0/16",  
    "Description": "vpc-a"  
  },  
  {  
    "Cidr": "10.2.0.0/16",  
    "Description": "vpc-b"  
  }  
]  
}
```

Per ulteriori informazioni, consulta [Managed Prefix Lists](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [GetManagedPrefixListEntries AWS CLI Command Reference](#).

## get-network-insights-access-scope-analysis-findings

Il seguente esempio di codice mostra come utilizzare `get-network-insights-access-scope-analysis-findings`.

### AWS CLI

Per ottenere i risultati di Network Insights, accedi all'analisi dell'ambito

L'`get-network-insights-access-scope-analysis-findings` esempio seguente ottiene i risultati dell'analisi dell'ambito selezionato nel tuo AWS account.

```
aws ec2 get-network-insights-access-scope-analysis-findings \  
  --region us-east-1 \  
  --network-insights-access-scope-analysis-id nisa \  
  --nis-123456789111
```

Output:

```
{  
  "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",  
  "AnalysisFindings": [  
    {  
      "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789222",  
      "NetworkInsightsAccessScopeId": "nis-123456789111",  
      "FindingComponents": [  
        {
```



- Per API i dettagli, vedere [GetNetworkInsightsAccessScopeAnalysisFindings](#) in AWS CLI Command Reference.

## get-network-insights-access-scope-content

Il seguente esempio di codice mostra come utilizzare `get-network-insights-access-scope-content`.

### AWS CLI

Per ottenere Network Insights, accedi ai contenuti dell'ambito

L'`get-network-insights-access-scope-content` esempio seguente ottiene il contenuto dell'ID di analisi dell'ambito selezionato nel tuo AWS account.

```
aws ec2 get-network-insights-access-scope-content \
  --region us-east-1 \
  --network-insights-access-scope-id nis-123456789222
```

Output:

```
{
  "NetworkInsightsAccessScopeContent": {
    "NetworkInsightsAccessScopeId": "nis-123456789222",
    "MatchPaths": [
      {
        "Source": {
          "ResourceStatement": {
            "ResourceTypes": [
              "AWS::EC2::NetworkInterface"
            ]
          }
        },
        "Destination": {
          "ResourceStatement": {
            "ResourceTypes": [
              "AWS::EC2::InternetGateway"
            ]
          }
        }
      }
    ]
  }
}
```

```
    ]
  }
}
```

Per ulteriori informazioni, vedere [Guida introduttiva all'utilizzo di Network Access Analyzer AWS CLI nella Guida di Network Access Analyzer](#).

- Per API i dettagli, vedere [GetNetworkInsightsAccessScopeContent](#) in AWS CLI Command Reference.

## get-password-data

Il seguente esempio di codice mostra come utilizzare `get-password-data`.

### AWS CLI

Per ottenere la password crittografata

In questo esempio viene ottenuta la password crittografata.

Comando:

```
aws ec2 get-password-data --instance-id i-1234567890abcdef0
```

Output:

```
{
  "InstanceId": "i-1234567890abcdef0",
  "Timestamp": "2013-08-07T22:18:38.000Z",
  "PasswordData": "gSlJFq+VpcZXqy+iktXMF6NyxQ4qCrT4+ga0uN0enX1MmgXPTj7XEXAMPLE
UQ+YeFfb+L1U4C4AKv652Ux1iRB3CPTY7WmU3TUnhsuBd+p6LVk7T2lKUm160Xbk6WPW1VYYm/TRPB1
e1DQ7PY4an/DgZT4mwcpRFigzhniQgDDe01InvSDcwoUTwNs0Y1S8ouri2W4n5GNlriM3Q0AnNVe1Vz/
53TkDtxbNoU606M1gK9zUWSxqEgwvbV2j8c5rP0WCuaMWSF14ziDu4bd7q+4RSyi8NUsVWnKZ4aEZffu
DPGzKrF5yL1f3etP2L4ZR6CvG7K1hx7VK0QVN32Dajw=="
}
```

Per ottenere la password decrittografata

In questo esempio viene ottenuta la password decrittografata.

Comando:



```
aws ec2 get-password-data --instance-id i-1234567890abcdef0 --priv-launch-key C:\Keys\MyKeyPair.pem
```

Output:

```
{
  "InstanceId": "i-1234567890abcdef0",
  "Timestamp": "2013-08-30T23:18:05.000Z",
  "PasswordData": "&ViJ652e*u"
}
```

- Per API i dettagli, vedere [GetPasswordData](#) in AWS CLI Command Reference.

## get-reserved-instances-exchange-quote

Il seguente esempio di codice mostra come utilizzare `get-reserved-instances-exchange-quote`.

AWS CLI

Per ottenere un preventivo per lo scambio di un'istanza riservata convertibile

Questo esempio ottiene le informazioni di scambio per le istanze riservate convertibili specificate.

Comando:

```
aws ec2 get-reserved-instances-exchange-quote --reserved-instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-configurations OfferingId=6fea5434-b379-434c-b07b-a7abexample
```

Output:

```
{
  "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"
  }
}

```

- Per API i dettagli, vedere [GetReservedInstancesExchangeQuote](#) in AWS CLI Command Reference.

## get-security-groups-for-vpc

Il seguente esempio di codice mostra come utilizzare `get-security-groups-for-vpc`.

### AWS CLI

Per visualizzare i gruppi di sicurezza che possono essere associati alle interfacce di rete in un determinato VPC periodo.

L'get-security-groups-for-vpc seguente mostra i gruppi di sicurezza che possono essere associati alle interfacce di rete in VPC

```
aws ec2 get-security-groups-for-vpc \  
  --vpc-id vpc-6c31a611 \  
  --region us-east-1
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [GetSecurityGroupsForVpc](#) in AWS CLI Command Reference.

## get-serial-console-access-status

Il seguente esempio di codice mostra come utilizzare get-serial-console-access-status.

### AWS CLI

Per visualizzare lo stato dell'account, accedere alla console seriale

L'get-serial-console-access-status seguente determina se l'accesso alla console seriale è abilitato per l'account.

```
aws ec2 get-serial-console-access-status
```

Output:

```
{
  "SerialConsoleAccessEnabled": true
}
```

Per ulteriori informazioni, consulta [EC2Serial Console](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [GetSerialConsoleAccessStatus AWS CLI Command Reference](#).

### **get-snapshot-block-public-access-state**

Il seguente esempio di codice mostra come utilizzare `get-snapshot-block-public-access-state`.

AWS CLI

Per ottenere lo stato attuale dell'accesso pubblico a blocchi per le istantanee

L'esempio seguente ottiene lo stato attuale dell'accesso pubblico a blocchi per le istantanee.

```
aws ec2 get-snapshot-block-public-access-state
```

Output:

```
{
  "State": "block-all-sharing"
}
```

Per ulteriori informazioni, consulta [Bloccare l'accesso pubblico per le istantanee](#) nella Amazon EBS User Guide.

- Per API i dettagli, consulta [GetSnapshotBlockPublicAccessState AWS CLI Command Reference](#).

### **get-spot-placement-scores**

Il seguente esempio di codice mostra come utilizzare `get-spot-placement-scores`.

## AWS CLI

Per calcolare il punteggio di posizionamento Spot per requisiti specifici

L'`get-spot-placement-scores` seguente genera innanzitutto un elenco di tutti i possibili parametri che possono essere specificati per la configurazione del punteggio di posizionamento Spot utilizzando il `--generate-cli-skeleton` parametro e salva l'elenco in un JSON file. Quindi, il JSON file viene utilizzato per configurare i requisiti da utilizzare per calcolare il punteggio di posizionamento Spot.

Per generare tutti i possibili parametri che possono essere specificati per la configurazione del punteggio di posizionamento Spot e salvare l'output direttamente in un JSON file.

```
aws ec2 get-spot-placement-scores \
  --region us-east-1 \
  --generate-cli-skeleton input > attributes.json
```

Output:

```
{
  "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": ""
}

```

Configura il file JSON. È necessario fornire un valore per `TargetCapacity`: Per una descrizione di ogni parametro e dei relativi valori predefiniti, consultate [Calculate the Spot placement score \(AWS CLI\)](https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/spotcalculate-sps-cli-placement-score.html#) <[https://docs.aws.amazon.com/AWS\\_EC2/latest/UserGuide/spotcalculate-sps-cli-placement-score.html#](https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/spotcalculate-sps-cli-placement-score.html#)>.

Calcola il punteggio di posizionamento Spot per i requisiti specificati in `attributes.json`. Specificate il nome e il percorso JSON del file utilizzando il `--cli-input-json` parametro.

```

aws ec2 get-spot-placement-scores \
  --region us-east-1 \
  --cli-input-json file://attributes.json

```

L'output if `SingleAvailabilityZone` è impostato su `false` o omesso (se omesso, il valore predefinito è). `false` Viene restituito un elenco di regioni con punteggio.

```

"Recommendation": [
  {
    "Region": "us-east-1",
    "Score": 7
  },
  {
    "Region": "us-west-1",
    "Score": 5
  },
  ...

```

L'output di `SingleAvailabilityZone` è impostato su `true`. Viene restituito un elenco di `SingleAvailabilityZone` con punteggio.

```
"Recommendation": [  
  {  
    "Region": "us-east-1",  
    "AvailabilityZoneId": "use1-az1"  
    "Score": 8  
  },  
  {  
    "Region": "us-east-1",  
    "AvailabilityZoneId": "usw2-az3"  
    "Score": 6  
  },  
  ...
```

Per ulteriori informazioni sul calcolo di un punteggio di posizionamento Spot e, ad esempio, sulle configurazioni, consulta [Calculate a Spot placement score](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetSpotPlacementScores](#) Reference.

## get-subnet-cidr-reservations

Il seguente esempio di codice mostra come utilizzare `get-subnet-cidr-reservations`.

### AWS CLI

Per ottenere informazioni su una prenotazione di sottorete CIDR

L'output di `get-subnet-cidr-reservations` seguente visualizza informazioni sulla prenotazione di sottorete CIDR specificata.

```
aws ec2 get-subnet-cidr-reservations \  
  --subnet-id subnet-03c51e2e6cEXAMPLE
```

Output:

```
{  
  "SubnetIpv4CidrReservations": [  
    {  
      "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
```



```

        "SubnetId": "subnet-03c51e2e6cEXAMPLE",
        "Cidr": "10.1.0.16/28",
        "ReservationType": "prefix",
        "OwnerId": "123456789012"
    }
],
"SubnetIpv6CidrReservations": []
}

```

Per ulteriori informazioni, consulta [Subnet CIDR reservations](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [GetSubnetCidrReservations AWS CLI Command Reference](#).

## get-transit-gateway-attachment-propagations

Il seguente esempio di codice mostra come utilizzare `get-transit-gateway-attachment-propagations`.

### AWS CLI

Per elencare le tabelle di routing a cui l'allegato di risorse specificato propaga le rotte

L'`get-transit-gateway-attachment-propagations` esempio seguente elenca la tabella delle rotte a cui l'allegato di risorse specificato propaga le rotte.

```

aws ec2 get-transit-gateway-attachment-propagations \
  --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE

```

Output:

```

{
  "TransitGatewayAttachmentPropagations": [
    {
      "TransitGatewayRouteTableId": "tgw-rtb-0882c61b97EXAMPLE",
      "State": "enabled"
    }
  ]
}

```

Per ulteriori informazioni, consulta le [tabelle delle rotte dei gateway di transito](#) nella Guida ai gateway di transito.

- Per API i dettagli, vedere [GetTransitGatewayAttachmentPropagations](#) in AWS CLI Command Reference.

## get-transit-gateway-multicast-domain-associations

Il seguente esempio di codice mostra come utilizzare `get-transit-gateway-multicast-domain-associations`.

### AWS CLI

Per visualizzare le informazioni sulle associazioni di domini multicast del gateway di transito

L'`get-transit-gateway-multicast-domain-associations` esempio seguente restituisce le associazioni per il dominio multicast specificato.

```
aws ec2 get-transit-gateway-multicast-domain-associations \
  --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

Output:

```
{
  "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"
    }
  },
  {
    "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
    "ResourceId": "vpc-7EXAMPLE",
    "ResourceType": "vpc",
    "Subnet": {
      "SubnetId": "subnet-fEXAMPLE",
      "State": "associated"
    }
  }
]
}

```

Per ulteriori informazioni, vedere [Managing multicast Domains](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [GetTransitGatewayMulticastDomainAssociations](#) in AWS CLI Command Reference.

## get-transit-gateway-prefix-list-references

Il seguente esempio di codice mostra come utilizzare `get-transit-gateway-prefix-list-references`.

### AWS CLI

Per ottenere i riferimenti all'elenco dei prefissi in una tabella di routing del gateway di transito

L'output di `aws ec2 get-transit-gateway-prefix-list-references` seguente ottiene i riferimenti all'elenco dei prefissi per la tabella di routing del gateway di transito specificata e filtra in base all'ID di un elenco di prefissi specifico.

```
aws ec2 get-transit-gateway-prefix-list-references \
  --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
  --filters Name=prefix-list-id,Values=pl-1111112222222333
```

Output:

```
{
  "TransitGatewayPrefixListReferences": [
    {
      "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
      "PrefixListId": "pl-1111112222222333",
      "PrefixListOwnerId": "123456789012",
      "State": "available",
      "Blackhole": false,
      "TransitGatewayAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
        "ResourceType": "vpc",
        "ResourceId": "vpc-112233445566aabbcc"
      }
    }
  ]
}
```

Per ulteriori informazioni, vedete i [riferimenti all'elenco dei prefissi](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [GetTransitGatewayPrefixListReferences](#) in AWS CLI Command Reference.

## get-transit-gateway-route-table-associations

Il seguente esempio di codice mostra come utilizzare `get-transit-gateway-route-table-associations`.

### AWS CLI

Per ottenere informazioni sulle associazioni per la tabella di routing del gateway di transito specificata

L'get-transit-gateway-route-table-associationsempio seguente visualizza informazioni sulle associazioni per la tabella di routing del gateway di transito specificata.

```
aws ec2 get-transit-gateway-route-table-associations \  
--transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE
```

Output:

```
{  
  "Associations": [  
    {  
      "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",  
      "ResourceId": "vpc-4d7de228",  
      "ResourceType": "vpc",  
      "State": "associating"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta le [tabelle di routing dei gateway](#) di transito nella Guida ai gateway di transito.

- Per API i dettagli, vedere [GetTransitGatewayRouteTableAssociations](#) in AWS CLI Command Reference.

## get-transit-gateway-route-table-propagations

Il seguente esempio di codice mostra come utilizzareget-transit-gateway-route-table-propagations.

AWS CLI

Per visualizzare informazioni sulle propagazioni della tabella di routing per la tabella di routing del gateway di transito specificata

L'get-transit-gateway-route-table-propagationsempio seguente restituisce le propagazioni della tabella di routing per la tabella di route specificata.

```
aws ec2 get-transit-gateway-route-table-propagations \  
--transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE
```

**Output:**

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta le [tabelle di routing del gateway Transit](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [GetTransitGatewayRouteTablePropagations](#) in AWS CLI Command Reference.

**get-verified-access-endpoint-policy**

Il seguente esempio di codice mostra come utilizzare `get-verified-access-endpoint-policy`.

**AWS CLI**

Per ottenere la politica di accesso verificato di un endpoint

L'`get-verified-access-endpoint-policy` esempio seguente ottiene la politica di accesso verificato dell'endpoint specificato.

```
aws ec2 get-verified-access-endpoint-policy \
```

```
--verified-access-endpoint-id vae-066fac616d4d546f2
```

Output:

```
{
  "PolicyEnabled": true,
  "PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
}
```

Per ulteriori informazioni, consulta le [politiche di accesso verificato](#) nella Guida per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [GetVerifiedAccessEndpointPolicy AWS CLI Command Reference](#).

## get-verified-access-group-policy

Il seguente esempio di codice mostra come utilizzare `get-verified-access-group-policy`.

AWS CLI

Per ottenere la politica di accesso verificato di un gruppo

L'`get-verified-access-group-policy` esempio seguente ottiene la politica di accesso verificato del gruppo specificato.

```
aws ec2 get-verified-access-group-policy \
  --verified-access-group-id vagr-0dbe967baf14b7235
```

Output:

```
{
  "PolicyEnabled": true,
  "PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
}
```

Per ulteriori informazioni, consulta [Gruppi di accesso verificato](#) nella Guida per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [GetVerifiedAccessGroupPolicy AWS CLI Command Reference](#).

## get-vpn-connection-device-sample-configuration

Il seguente esempio di codice mostra come utilizzare `get-vpn-connection-device-sample-configuration`.

### AWS CLI

Per scaricare un file di configurazione di esempio

L'`get-vpn-connection-device-sample-configuration` seguente scarica il file di configurazione di esempio specificato. Per elencare i dispositivi gateway con un file di configurazione di esempio, chiamate il `get-vpn-connection-device-types` comando.

```
aws ec2 get-vpn-connection-device-sample-configuration \
  --vpn-connection-id vpn-123456789abc01234 \
  --vpn-connection-device-type-id 5fb390ba
```

Output:

```
{
  "VpnConnectionDeviceSampleConfiguration": "contents-of-the-sample-configuration-
file"
}
```

Per ulteriori informazioni, consulta [Scaricare il file di configurazione](#) nella Guida AWS Site-to-Site VPN per l'utente.

- Per API i dettagli, vedere [GetVpnConnectionDeviceSampleConfiguration](#) in AWS CLI Command Reference.

## get-vpn-connection-device-types

Il seguente esempio di codice mostra come utilizzare `get-vpn-connection-device-types`.

### AWS CLI

Per elencare i dispositivi gateway con un file di configurazione di esempio



L'output di `get-vpn-connection-device-types` seguente elenca i dispositivi gateway di Palo Alto Networks che dispongono di file di configurazione di esempio.

```
aws ec2 get-vpn-connection-device-types \
  --query "VpnConnectionDeviceTypes[?Vendor=='Palo Alto Networks']"
```

Output:

```
[
  {
    "VpnConnectionDeviceTypeId": "754a6372",
    "Vendor": "Palo Alto Networks",
    "Platform": "PA Series",
    "Software": "PANOS 4.1.2+"
  },
  {
    "VpnConnectionDeviceTypeId": "9612cbcd",
    "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+"
  }
]
```

Per ulteriori informazioni, vedere [Scaricare il file di configurazione](#) nella Guida per l'AWS Site-to-Site VPN utente.

- Per API i dettagli, vedere [GetVpnConnectionDeviceTypes](#) in AWS CLI Command Reference.

## **import-client-vpn-client-certificate-revocation-list**

Il seguente esempio di codice mostra come utilizzare `import-client-vpn-client-certificate-revocation-list`.

AWS CLI

Per importare un elenco di revoche di certificati client

L'import-client-vpn-client-certificate-revocation-list esempio seguente importa un elenco di revoca dei certificati client nell'VPNEndpoint Client specificando la posizione del file sul computer locale.

```
aws ec2 import-client-vpn-client-certificate-revocation-list \  
  --certificate-revocation-list file:///path/to/crl.pem \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

Output:

```
{  
  "Return": true  
}
```

Per ulteriori informazioni, vedere [Elenchi di revoca dei certificati client](#) nella Client Administrator Guide.AWS VPN

- Per API i dettagli, vedere [ImportClientVpnClientCertificateRevocationList](#) in AWS CLI Command Reference.

## import-image

Il seguente esempio di codice mostra come utilizzare import-image.

### AWS CLI

Per importare un file di immagine VM come AMI

L'import-image esempio seguente importa quanto specificato OVA.

```
aws ec2 import-image \  
  --disk-containers Format=ova,UserBucket="{S3Bucket=my-import-bucket,S3Key=vms/my-server-vm.ova}"
```

Output:

```
{  
  "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"  
}
```

- Per API i dettagli, vedere [ImportImage](#) in AWS CLI Command Reference.

## import-key-pair

Il seguente esempio di codice mostra come utilizzare `import-key-pair`.

### AWS CLI

Per importare una chiave pubblica

Innanzitutto, genera una key pair con lo strumento che preferisci. Ad esempio, usa questo comando `ssh-keygen`:

Comando:

```
ssh-keygen -t rsa -C "my-key" -f ~/.ssh/my-key
```

Output:

```
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.  
...
```

Questo comando di esempio importa la chiave pubblica specificata.

Comando:

```
aws ec2 import-key-pair --key-name "my-key" --public-key-material fileb://~/.ssh/my-key.pub
```

Output:

```
{
  "KeyName": "my-key",
  "KeyFingerprint": "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca"
}
```

- Per API i dettagli, vedere [ImportKeyPair](#) in AWS CLI Command Reference.

## import-snapshot

Il seguente esempio di codice mostra come utilizzare `import-snapshot`.

### AWS CLI

Per importare un'istantanea

L'`import-snapshot` esempio seguente importa il disco specificato come istantanea.

```
aws ec2 import-snapshot \
  --description "My server VMDK" \
  --disk-container Format=VMDK,UserBucket={S3Bucket=my-import-bucket,S3Key=vms/my-server-vm.vmdk}
```

Output:

```
{
  "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"
      }
    }
  }
```

- Per API i dettagli, vedere [ImportSnapshot](#) in AWS CLI Command Reference.

## list-images-in-recycle-bin

Il seguente esempio di codice mostra come utilizzare `list-images-in-recycle-bin`.

### AWS CLI

Per elencare le immagini nel Cestino

L'`list-images-in-recycle-bin` seguente elenca tutte le immagini attualmente conservate nel Cestino.

```
aws ec2 list-images-in-recycle-bin
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Recover AMIs from the Recycle Bin](#) nella Amazon Elastic Compute Cloud User Guide.

- Per API i dettagli, consulta AWS CLI Command [ListImagesInRecycleBin](#) Reference.

## list-snapshots-in-recycle-bin

Il seguente esempio di codice mostra come utilizzare `list-snapshots-in-recycle-bin`.

## AWS CLI

Per visualizzare le istantanee nel Cestino

L'`list-snapshots-in-recycle-bin` seguente elenca le informazioni sulle istantanee nel Cestino, tra cui l'ID dell'istananea, una descrizione dell'istananea, l'ID del volume da cui è stata creata l'istananea, la data e l'ora in cui l'istananea è stata eliminata ed è stata inserita nel Cestino e la data e l'ora di scadenza del periodo di conservazione.

```
aws ec2 list-snapshots-in-recycle-bin \
  --snapshot-id snap-01234567890abcdef
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni su Recycle Bin for AmazonEBS, consulta [Recover snapshot from the Recycle Bin](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta Command [ListSnapshotsInRecycleBin](#)Reference AWS CLI .

## lock-snapshot

Il seguente esempio di codice mostra come utilizzare `lock-snapshot`.

### AWS CLI

Esempio 1: bloccare un'istananea in modalità di governance

L'`lock-snapshot` seguente blocca l'istananea specificata in modalità di governance.

```
aws ec2 lock-snapshot \
```

```
--snapshot-id snap-0b5e733b4a8df6e0d \  
--lock-mode governance \  
--lock-duration 365
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Snapshot lock](#) nella Amazon EBS User Guide.

Esempio 2: per bloccare uno snapshot in modalità di conformità

L'lock-snapshotesempio seguente blocca l'istantanea specificata in modalità di conformità.

```
aws ec2 lock-snapshot \  
--snapshot-id snap-0163a8524c5b9901f \  
--lock-mode compliance \  
--cool-off-period 24 \  
--lock-duration 365
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Snapshot lock](#) nella Amazon EBS User Guide.

- Per API i dettagli, consulta [LockSnapshot AWS CLI Command Reference](#).

## modify-address-attribute

Il seguente esempio di codice mostra come utilizzare `modify-address-attribute`.

### AWS CLI

Per modificare l'attributo del nome di dominio associato a un indirizzo IP elastico

`modify-address-attribute` Gli esempi seguenti modificano l'attributo del nome di dominio di un indirizzo IP elastico.

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
```

Output:

```
{  
  "Addresses": [  
    {  
      "PublicIp": "192.0.2.0",  
      "AllocationId": "eipalloc-abcdef01234567890",  
      "PtrRecord": "example.net."  
      "PtrRecordUpdate": {  
        "Value": "example.com.",  
        "Status": "PENDING"  
      }  
    }  
  ]  
}
```

Per monitorare la modifica in sospeso e per visualizzare gli attributi modificati di un indirizzo IP elastico, vedere [describe-addresses-attribute](#) nel AWS CLI Command Reference.

- Per API i dettagli, vedere [ModifyAddressAttribute](#) in AWS CLI Command Reference.



## modify-availability-zone-group

Il seguente esempio di codice mostra come utilizzare `modify-availability-zone-group`.

### AWS CLI

Per abilitare un gruppo di zone

L'esempio seguente abilita il gruppo di zone specificato.

```
aws ec2 modify-availability-zone-group \  
  --group-name us-west-2-lax-1 \  
  --opt-in-status opted-in
```

Output:

```
{  
  "Return": true  
}
```

Per ulteriori informazioni, consulta [Regioni e zone](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per le istanze Linux.

- Per API i dettagli, consulta AWS CLI Command [ModifyAvailabilityZoneGroupReference](#).

## modify-capacity-reservation-fleet

Il seguente esempio di codice mostra come utilizzare `modify-capacity-reservation-fleet`.

### AWS CLI

Esempio 1: modificare la capacità target totale di una flotta di prenotazioni di capacità

L'esempio seguente modifica la capacità target totale del parco veicoli di prenotazione di capacità specificato. Quando modifichi la capacità target totale di un parco istanze di prenotazione della capacità, questo crea automaticamente nuove prenotazioni di capacità o modifica o annulla le prenotazioni di capacità esistenti nel parco istanze per soddisfare la nuova capacità target totale. Non è possibile tentare ulteriori modifiche a un parco istanze mentre si trova nello stato `modifying`.

```
aws ec2 modify-capacity-reservation-fleet \  
  --fleet-id fleet-id \  
  --target-capacity target-capacity
```

```
--capacity-reservation-fleet-id crf-01234567890abcdef \  
--total-target-capacity 160
```

Output:

```
{  
  "Return": true  
}
```

Esempio 2: modificare la data di fine di un parco veicoli di riserva di capacità

L'`modify-capacity-reservation-fleet` seguente modifica la data di fine della flotta di prenotazioni di capacità specificata. Quando modifichi la data di fine del parco istanze, le date di fine per tutte le singole prenotazioni di capacità vengono aggiornate di conseguenza. Non è possibile tentare ulteriori modifiche a un parco istanze mentre si trova nello stato `modifying`.

```
aws ec2 modify-capacity-reservation-fleet \  
  --capacity-reservation-fleet-id crf-01234567890abcdef \  
  --end-date 2022-07-04T23:59:59.000Z
```

Output:

```
{  
  "Return": true  
}
```

Per ulteriori informazioni sulle flotte di prenotazione della capacità, consulta [Flotte di prenotazione della capacità nella Guida](#) per gli EC2 utenti di Amazon.

- Per API i dettagli, consulta AWS CLI Command [ModifyCapacityReservationFleet](#) Reference.

## **modify-capacity-reservation**

Il seguente esempio di codice mostra come utilizzare `modify-capacity-reservation`.

### AWS CLI

Esempio 1: per modificare il numero di istanze riservate da una prenotazione di capacità esistente

L'`modify-capacity-reservation` seguente modifica il numero di istanze per le quali la prenotazione di capacità riserva capacità.

```
aws ec2 modify-capacity-reservation \  
  --capacity-reservation-id cr-1234abcd56EXAMPLE \  
  --instance-count 5
```

Output:

```
{  
  "Return": true  
}
```

Esempio 2: modificare la data e l'ora di fine di una prenotazione di capacità esistente

L'`modify-capacity-reservation` esempio seguente modifica una prenotazione di capacità esistente in modo che termini alla data e all'ora specificate.

```
aws ec2 modify-capacity-reservation \  
  --capacity-reservation-id cr-1234abcd56EXAMPLE \  
  --end-date-type Limited \  
  --end-date 2019-08-31T23:59:59Z
```

Per ulteriori informazioni, consulta [Modificare una prenotazione di capacità](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per API i dettagli, consulta Command [ModifyCapacityReservation](#) Reference AWS CLI .

## modify-client-vpn-endpoint

Il seguente esempio di codice mostra come utilizzare `modify-client-vpn-endpoint`.

AWS CLI

Per modificare un VPN endpoint Client

L'`modify-client-vpn-endpoint` esempio seguente abilita la registrazione della connessione client per l'endpoint Client VPN specificato.

```
aws ec2 modify-client-vpn-endpoint \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --connection-log-options Enabled=true,CloudwatchLogGroup=ClientVPNLogs
```

Output:

```
{
  "Return": true
}
```

Per ulteriori informazioni, vedere [Client VPN Endpoints nella Client Administrator AWS Guide](#).  
VPN

- Per API i dettagli, vedere [ModifyClientVpnEndpoint](#) in AWS CLI Command Reference.

## **modify-default-credit-specification**

Il seguente esempio di codice mostra come utilizzare `modify-default-credit-specification`.

AWS CLI

Per modificare l'opzione di credito predefinita

L'esempio seguente modifica l'opzione di credito predefinita per le istanze T2.

```
aws ec2 modify-default-credit-specification \
  --instance-family t2 \
  --cpu-credits unlimited
```

Output:

```
{
  "InstanceFamilyCreditSpecification": {
    "InstanceFamily": "t2",
    "CpuCredits": "unlimited"
  }
}
```

- Per API i dettagli, vedere [ModifyDefaultCreditSpecification](#) in AWS CLI Command Reference.

## **modify-ebs-default-kms-key-id**

Il seguente esempio di codice mostra come utilizzare `modify-ebs-default-kms-key-id`.

## AWS CLI

Per impostare l'impostazione predefinita CMK per la EBS crittografia

L'`modify-ebs-default-kms-key-id` seguente imposta CMK quanto specificato come predefinito CMK per la EBS crittografia per l' AWS account nella regione corrente.

```
aws ec2 modify-ebs-default-kms-key-id \  
  --kms-key-id alias/my-cmk
```

Output:

```
{  
  "KmsKeyId": "arn:aws:kms:us-  
west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"  
}
```

- Per API i dettagli, vedere [ModifyEbsDefaultKmsKeyId](#) in AWS CLI Command Reference.

## `modify-fleet`

Il seguente esempio di codice mostra come utilizzare `modify-fleet`.

## AWS CLI

Per scalare una EC2 flotta

L'`modify-fleet` seguente modifica la capacità target della EC2 flotta specificata. Se il valore specificato è maggiore della capacità corrente, EC2 Fleet avvia istanze aggiuntive. Se il valore specificato è inferiore alla capacità corrente, il EC2 parco istanze annulla tutte le richieste aperte e, se la politica di terminazione lo è `terminate`, termina tutte le EC2 istanze che superano la nuova capacità target.

```
aws ec2 modify-fleet \  
  --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \  
  --target-capacity-specification TotalTargetCapacity=5
```

Output:

```
{  
  "Return": true
```

```
}
```

Per ulteriori informazioni, consulta [Managing an EC2 Fleet](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

- Per API i dettagli, consulta AWS CLI Command [ModifyFleetReference](#).

## modify-fpga-image-attribute

Il seguente esempio di codice mostra come utilizzare `modify-fpga-image-attribute`.

### AWS CLI

Per modificare gli attributi di un'FPGA immagine Amazon

Questo esempio aggiunge le autorizzazioni di caricamento per l'ID dell'account 123456789012 per quanto specificato AFID.

Comando:

```
aws ec2 modify-fpga-image-attribute --attribute LoadPermission --fpga-image-id afi-0d123e123bfc85abc --load-permission Add=[{UserId=123456789012}]
```

Output:

```
{
  "FpgaImageAttribute": {
    "FpgaImageId": "afi-0d123e123bfc85abc",
    "LoadPermissions": [
      {
        "UserId": "123456789012"
      }
    ]
  }
}
```

- Per API i dettagli, vedere [ModifyFpgaImageAttribute](#) in AWS CLI Command Reference.

## modify-hosts

Il seguente esempio di codice mostra come utilizzare `modify-hosts`.

## AWS CLI

Esempio 1: abilitare il posizionamento automatico per un host dedicato

L'`modify-hosts` esempio seguente abilita il posizionamento automatico per un host dedicato in modo che accetti qualsiasi avvio di istanza non mirato che corrisponda alla configurazione del tipo di istanza.

```
aws ec2 modify-hosts \  
  --host-id h-06c2f189b4EXAMPLE \  
  --auto-placement on
```

Output:

```
{  
  "Successful": [  
    "h-06c2f189b4EXAMPLE"  
  ],  
  "Unsuccessful": []  
}
```

Esempio 2: abilitare il ripristino dell'host per un host dedicato

L'`modify-hosts` esempio seguente abilita il ripristino dell'host per l'host dedicato specificato.

```
aws ec2 modify-hosts \  
  --host-id h-06c2f189b4EXAMPLE \  
  --host-recovery on
```

Output:

```
{  
  "Successful": [  
    "h-06c2f189b4EXAMPLE"  
  ],  
  "Unsuccessful": []  
}
```

Per ulteriori informazioni, consulta [Modificare il posizionamento automatico dell'host dedicato](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per i API dettagli, consulta [ModifyHosts](#) Command Reference.AWS CLI

## modify-id-format

Il seguente esempio di codice mostra come utilizzare `modify-id-format`.

### AWS CLI

Per abilitare il formato ID più lungo per una risorsa

L'`modify-id-format` esempio seguente abilita il formato ID più lungo per il tipo di `instance` risorsa.

```
aws ec2 modify-id-format \  
  --resource instance \  
  --use-long-ids
```

Per disabilitare il formato ID più lungo per una risorsa

L'`modify-id-format` esempio seguente disattiva il formato ID più lungo per il tipo di `instance` risorsa.

```
aws ec2 modify-id-format \  
  --resource instance \  
  --no-use-long-ids
```

L'`modify-id-format` esempio seguente abilita il formato ID più lungo per tutti i tipi di risorse supportati che rientrano nel periodo di attivazione.

```
aws ec2 modify-id-format \  
  --resource all-current \  
  --use-long-ids
```

- Per API i dettagli, vedere [ModifyIdFormat](#) in AWS CLI Command Reference.

## modify-identity-id-format

Il seguente esempio di codice mostra come utilizzare `modify-identity-id-format`.

### AWS CLI

Per consentire un utilizzo prolungato di un IAM ruolo IDs per una risorsa



L'`modify-identity-id-format` seguente consente al IAM ruolo `EC2Role` nell' AWS account di utilizzare il formato ID lungo per il tipo di `instance` risorsa.

```
aws ec2 modify-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:role/EC2Role \  
  --resource instance \  
  --use-long-ids
```

Per consentire a un IAM utente di utilizzare più a IDs lungo una risorsa

L'`modify-identity-id-format` seguente consente `AdminUser` all'IAM utente del tuo AWS account di utilizzare il formato ID più lungo per il tipo di `volume` risorsa.

```
aws ec2 modify-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:user/AdminUser \  
  --resource volume \  
  --use-long-ids
```

L'`modify-identity-id-format` seguente consente IAM `AdminUser` all'utente del tuo AWS account di utilizzare il formato ID più lungo per tutti i tipi di risorse supportati che rientrano nel periodo di attivazione.

```
aws ec2 modify-identity-id-format \  
  --principal-arn arn:aws:iam::123456789012:user/AdminUser \  
  --resource all-current \  
  --use-long-ids
```

- Per API i dettagli, vedere [ModifyIdentityIdFormat](#) in AWS CLI Command Reference.

## **modify-image-attribute**

Il seguente esempio di codice mostra come utilizzare `modify-image-attribute`.

### AWS CLI

Esempio 1: rendere AMI pubblico un

L'`modify-instance-attribute` seguente rende AMI pubblico quanto specificato.

```
aws ec2 modify-image-attribute \  
  --use-long-ids
```

```
--image-id ami-5731123e \  
--launch-permission "Add=[{Group=all}]"
```

Questo comando non produce alcun output.

Esempio 2: renderlo AMI privato

L'`modify-instance-attribute` esempio seguente rende AMI privato il dato specificato.

```
aws ec2 modify-image-attribute \  
--image-id ami-5731123e \  
--launch-permission "Remove=[{Group=all}]"
```

Questo comando non produce alcun output.

Esempio 3: concedere l'autorizzazione di avvio a un AWS account

L'`modify-instance-attribute` esempio seguente concede le autorizzazioni di avvio all'account specificato AWS .

```
aws ec2 modify-image-attribute \  
--image-id ami-5731123e \  
--launch-permission "Add=[{UserId=123456789012}]"
```

Questo comando non produce alcun output.

Esempio 4: Per rimuovere l'autorizzazione di avvio da un account AWS

L'`modify-instance-attribute` esempio seguente rimuove le autorizzazioni di avvio dall' AWS account specificato.

```
aws ec2 modify-image-attribute \  
--image-id ami-5731123e \  
--launch-permission "Remove=[{UserId=123456789012}]"
```

- Per API i dettagli, vedere [ModifyImageAttribute](#) in AWS CLI Command Reference.

## **modify-instance-attribute**

Il seguente esempio di codice mostra come utilizzare `modify-instance-attribute`.

## AWS CLI

Esempio 1: per modificare il tipo di istanza

L'`modify-instance-attribute` seguente modifica il tipo di istanza dell'istanza specificata. L'istanza deve essere nello stato `stopped`.

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --instance-type "{\"Value\": \"m1.small\"}"
```

Questo comando non produce alcun output.

Esempio 2: abilitare una rete avanzata su un'istanza

L'`modify-instance-attribute` seguente abilita una rete avanzata per l'istanza specificata. L'istanza deve essere nello stato `stopped`.

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --sriov-net-support simple
```

Questo comando non produce alcun output.

Esempio 3: Per modificare l' `sourceDestCheck` attributo

L'`modify-instance-attribute` seguente imposta l'`sourceDestCheck` attributo dell'istanza specificata su `true`. L'istanza deve trovarsi in un VPC.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --source-dest-  
check "{\"Value\": true}"
```

Questo comando non produce alcun output.

Esempio 4: modificare l' `deleteOnTermination` attributo del volume principale

L'`modify-instance-attribute` seguente imposta l'`deleteOnTermination` attributo per il volume root dell'istanza EBS supportata da Amazon specificata su `false`. Per impostazione predefinita, questo attributo è `true` per il volume root.

Comando:

```
aws ec2 modify-instance-attribute \  
  --instance-id i-1234567890abcdef0 \  
  --block-device-mappings "[{\"DeviceName\": \"/dev/sda1\", \"Ebs\":  
{\"DeleteOnTermination\": false}}]"
```

Questo comando non produce alcun output.

Esempio 5: modificare i dati utente allegati a un'istanza

L'`modify-instance-attribute` esempio seguente aggiunge il contenuto del file `UserData.txt` come `UserData` per l'istanza specificata.

Contenuto del file originale `UserData.txt`:

```
#!/bin/bash  
yum update -y  
service httpd start  
chkconfig httpd on
```

Il contenuto del file deve essere codificato in base64. Il primo comando converte il file di testo in base64 e lo salva come nuovo file.

Versione Linux/macOS del comando:

```
base64 UserData.txt > UserData.base64.txt
```

Questo comando non produce alcun output.

Versione Windows del comando:

```
certutil -encode UserData.txt tmp.b64 && findstr /v /c:- tmp.b64 >  
UserData.base64.txt
```

Output:

```
Input Length = 67  
Output Length = 152  
CertUtil: -encode command completed successfully.
```

Ora puoi fare riferimento a quel file nel CLI comando che segue:

```
aws ec2 modify-instance-attribute \  
  --instance-id=i-09b5a14dbca622e76 \  
  --attribute userData --value file://UserData.base64.txt
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Dati utente e AWS CLI nella Guida per l'EC2utente](#).

- Per API i dettagli, vedere [ModifyInstanceAttribute](#) in AWS CLI Command Reference.

## modify-instance-capacity-reservation-attributes

Il seguente esempio di codice mostra come utilizzare `modify-instance-capacity-reservation-attributes`.

### AWS CLI

Esempio 1: modificare le impostazioni di targeting per la prenotazione della capacità di un'istanza

L'`modify-instance-capacity-reservation-attributes` esempio seguente modifica un'istanza interrotta per indirizzare una prenotazione di capacità specifica.

```
aws ec2 modify-instance-capacity-reservation-attributes \  
  --instance-id i-EXAMPLE8765abcd4e \  
  --capacity-reservation-specification  
  'CapacityReservationTarget={CapacityReservationId= cr-1234abcd56EXAMPLE }'
```

Output:

```
{  
  "Return": true  
}
```

Esempio 2: modificare le impostazioni di targeting per la prenotazione della capacità di un'istanza

L'`modify-instance-capacity-reservation-attributes` esempio seguente modifica un'istanza interrotta che ha come destinazione la prenotazione di capacità specificata per avviarla in qualsiasi prenotazione di capacità che abbia attributi corrispondenti (tipo di istanza, piattaforma, zona di disponibilità) e che abbia criteri di corrispondenza delle istanze aperte.

```
aws ec2 modify-instance-capacity-reservation-attributes \  
  --instance-id i-EXAMPLE8765abcd4e \  
  --capacity-reservation-specification  
  'CapacityReservationTarget={CapacityReservationId= cr-1234abcd56EXAMPLE }'
```

```
--instance-id i-EXAMPLE8765abcd4e \  
--capacity-reservation-specification 'CapacityReservationPreference=open'
```

Output:

```
{  
  "Return": true  
}
```

Per ulteriori informazioni, consulta [Modifica delle impostazioni di prenotazione della capacità di un'istanza](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per API i dettagli, consulta Command [ModifyInstanceCapacityReservationAttributesReference](#) AWS CLI .

## modify-instance-credit-specification

Il seguente esempio di codice mostra come utilizzare `modify-instance-credit-specification`.

AWS CLI

Per modificare l'opzione di credito per l'CPU utilizzo di un'istanza

Questo esempio modifica l'opzione di credito per l'CPU utilizzo dell'istanza specificata nella regione specificata in «unlimited». Le opzioni di credito valide sono «standard» e «illimitate».

Comando:

```
aws ec2 modify-instance-credit-specification --instance-credit-  
specification "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

Output:

```
{  
  "SuccessfulInstanceCreditSpecifications": [  
    {  
      "InstanceId": "i-1234567890abcdef0"  
    }  
  ],  
}
```

```
"UnsuccessfulInstanceCreditSpecifications": []
}
```

- Per API i dettagli, vedere [ModifyInstanceCreditSpecification](#) in AWS CLI Command Reference.

## modify-instance-event-start-time

Il seguente esempio di codice mostra come utilizzare `modify-instance-event-start-time`.

### AWS CLI

Per modificare l'ora di inizio dell'evento per un'istanza

Il `modify-instance-event-start-time` comando seguente mostra come modificare l'ora di inizio dell'evento per l'istanza specificata. Specificate l'ID dell'evento utilizzando il `--instance-event-id` parametro. Specificate la nuova data e ora utilizzando il `--not-before` parametro.

```
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
```

Output:

```
"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"
}
```

Per ulteriori informazioni, consulta [Lavorare con le istanze pianificate per il riavvio](#) nella Guida per l'utente di Amazon Elastic Compute Cloud

- Per API i dettagli, consulta Command [ModifyInstanceEventStartTime](#) Reference AWS CLI .

## modify-instance-event-window

Il seguente esempio di codice mostra come utilizzare `modify-instance-event-window`.

## AWS CLI

### Esempio 1: modificare l'intervallo di tempo di una finestra di evento

L'`modify-instance-event-window` seguente modifica l'intervallo di tempo di una finestra di eventi. Specifica il parametro `time-range` per modificare l'intervallo di tempo. Non è possibile specificare anche il parametro `cron-expression`.

```
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
```

Output:

```
{  
  "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"  
      }  
    ]  
  }  
}
```



```
}
```

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella sezione Eventi pianificati della Amazon EC2 User Guide.

Esempio 2: modificare una serie di intervalli di tempo per una finestra di evento

L'`modify-instance-event-window` seguente modifica l'intervallo di tempo di una finestra di eventi. Specifica il parametro `time-range` per modificare l'intervallo di tempo. Non è possibile specificare anche il parametro `cron-expression`.

```
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}]'
```

Output:

```
{
  "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"
    }
  ]
}
}

```

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella sezione Eventi pianificati della Amazon EC2 User Guide.

Esempio 3: modificare l'espressione cron di una finestra di evento

L'`modify-instance-event-window` seguente modifica l'espressione cron di una finestra di eventi. Specifica il parametro `cron-expression` per modificare l'espressione cron. Non è possibile specificare anche il parametro `time-range`.

```

aws ec2 modify-instance-event-window \
  --region us-east-1 \
  --instance-event-window-id iew-0abcdef1234567890 \
  --cron-expression "* 21-23 * * 2,3"

```

Output:

```

{
  "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"
        }
    ]
}
}

```

Per i vincoli relativi alla finestra degli eventi, consulta [Considerazioni](#) nella sezione Eventi pianificati della Amazon EC2 User Guide.

- Per API i dettagli, consulta Command [ModifyInstanceEventWindow](#)Reference AWS CLI .

## modify-instance-maintenance-options

Il seguente esempio di codice mostra come utilizzare `modify-instance-maintenance-options`.

AWS CLI

Esempio 1: disabilitare il comportamento di ripristino di un'istanza

L'`modify-instance-maintenance-options`esempio seguente disabilita il ripristino automatico semplificato per un'istanza in esecuzione o interrotta.

```

aws ec2 modify-instance-maintenance-options \
  --instance-id i-0abcdef1234567890 \
  --auto-recovery disabled

```

Output:

```

{
  "InstanceId": "i-0abcdef1234567890",
  "AutoRecovery": "disabled"
}

```

Per ulteriori informazioni, consulta [Recover your instance](#) nella Amazon EC2 User Guide for Linux Instances.

Esempio 2: per impostare come predefinito il comportamento di ripristino di un'istanza

L'`modify-instance-maintenance-options` seguente imposta il comportamento di ripristino automatico come predefinito, il che consente il ripristino automatico semplificato per i tipi di istanze supportati.

```
aws ec2 modify-instance-maintenance-options \  
  --instance-id i-0abcdef1234567890 \  
  --auto-recovery default
```

Output:

```
{  
  "InstanceId": "i-0abcdef1234567890",  
  "AutoRecovery": "default"  
}
```

Per ulteriori informazioni, consulta [Recover your instance](#) nella Amazon EC2 User Guide for Linux Instances.

- Per API i dettagli, consulta [ModifyInstanceMaintenanceOptions AWS CLI](#) Command Reference.

## `modify-instance-metadata-options`

Il seguente esempio di codice mostra come utilizzare `modify-instance-metadata-options`.

AWS CLI

Esempio 1: abilitare IMDSv2

L'`modify-instance-metadata-options` seguente configura l'uso di IMDSv2 sull'istanza specificata.

```
aws ec2 modify-instance-metadata-options \  
  --instance-id i-1234567898abcdef0 \  
  --http-tokens required \  
  --http-endpoint enabled
```

Output:

```
{
```

```
"InstanceId": "i-1234567898abcdef0",
"InstanceMetadataOptions": {
  "State": "pending",
  "HttpTokens": "required",
  "HttpPutResponseHopLimit": 1,
  "HttpEndpoint": "enabled"
}
}
```

Per ulteriori informazioni, consulta [Metadati dell'istanza e dati utente](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

Esempio 2: per disabilitare i metadati delle istanze

L'`modify-instance-metadata-options` seguente disabilita l'uso di tutte le versioni dei metadati dell'istanza sull'istanza specificata.

```
aws ec2 modify-instance-metadata-options \
  --instance-id i-1234567898abcdef0 \
  --http-endpoint disabled
```

Output:

```
{
  "InstanceId": "i-1234567898abcdef0",
  "InstanceMetadataOptions": {
    "State": "pending",
    "HttpTokens": "required",
    "HttpPutResponseHopLimit": 1,
    "HttpEndpoint": "disabled"
  }
}
```

Per ulteriori informazioni, consulta [Metadati dell'istanza e dati utente](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

Esempio 3: per abilitare l'endpoint dei metadati IPv6 dell'istanza per la tua istanza

L'`modify-instance-metadata-options` seguente mostra come attivare l'IPv6endpoint per il servizio di metadati dell'istanza.

```
aws ec2 modify-instance-metadata-options \
```

```
--instance-id i-1234567898abcdef0 \  
--http-protocol-ipv6 enabled \  
--http-endpoint enabled
```

Output:

```
{  
  "InstanceId": "i-1234567898abcdef0",  
  "InstanceMetadataOptions": {  
    "State": "pending",  
    "HttpTokens": "required",  
    "HttpPutResponseHopLimit": 1,  
    "HttpEndpoint": "enabled",  
    "HttpProtocolIpv6": "enabled"  
  }  
}
```

Per impostazione predefinita, l'IPv6endpoint è disabilitato. Questo è vero anche se hai avviato un'istanza in una sottorete IPv6 solo. L'IPv6endpoint per IMDS è accessibile solo sulle istanze create sul sistema Nitro. Per ulteriori informazioni, consulta [Metadati dell'istanza e dati utente](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per API i dettagli, consulta Command [ModifyInstanceMetadataOptions](#)Reference AWS CLI .

## modify-instance-placement

Il seguente esempio di codice mostra come utilizzare `modify-instance-placement`.

AWS CLI

Esempio 1: rimuovere l'affinità di un'istanza con un host dedicato

L'esempio seguente rimuove l'affinità di un'istanza con un host dedicato e ne consente l'avvio su qualsiasi host dedicato disponibile nell'account che supporti il tipo di istanza corrispondente.

```
aws ec2 modify-instance-placement \  
--instance-id i-0e6ddf6187EXAMPLE \  
--affinity default
```

Output:

```
{
  "Return": true
}
```

Esempio 2: stabilire l'affinità tra un'istanza e l'host dedicato specificato

L'`modify-instance-placement` seguente stabilisce una relazione di avvio tra un'istanza e un host dedicato. L'istanza può essere eseguita solo sull'host dedicato specificato.

```
aws ec2 modify-instance-placement \
  --instance-id i-0e6ddf6187EXAMPLE \
  --affinity host \
  --host-id i-0e6ddf6187EXAMPLE
```

Output:

```
{
  "Return": true
}
```

Per ulteriori informazioni, consulta [Modifying Instance Tenancy and Affinity](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.

Esempio 3: spostare un'istanza in un gruppo di collocamento

L'`modify-instance-placement` seguente sposta un'istanza in un gruppo di posizionamenti, interrompe l'istanza, modifica il posizionamento dell'istanza e quindi riavvia l'istanza.

```
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
```

Per ulteriori informazioni, consulta [Changing the Placement Group for an Instance](#) nella Amazon Elastic Compute Cloud Users Guide.

## Esempio 4: rimuovere un'istanza da un gruppo di collocamento

L'`modify-instance-placement` seguente rimuove un'istanza da un gruppo di posizionamenti interrompendo l'istanza, modificando il posizionamento dell'istanza e quindi riavviando l'istanza. L'esempio seguente specifica una stringa vuota («») per il nome del gruppo di posizionamento per indicare che l'istanza non deve essere posizionata in un gruppo di posizionamento.

Arresta l'istanza:

```
aws ec2 stop-instances \  
  --instance-ids i-0123a456700123456
```

Modifica la posizione (Windows Command Prompt, Linux e macOS):

```
aws ec2 modify-instance-placement \  
  --instance-id i-0123a456700123456 \  
  --group-name ""
```

Modifica la posizione (Windows PowerShell):

```
aws ec2 modify-instance-placement `\  
  --instance-id i-0123a456700123456 `\  
  --group-name ""
```

Riavvia l'istanza:

```
aws ec2 start-instances \  
  --instance-ids i-0123a456700123456
```

Output:

```
{  
  "Return": true  
}
```

Per ulteriori informazioni, consulta [Modifying Instance Tenancy and Affinity](#) nella Amazon Elastic Compute Cloud User Guide for Linux Instances.



- Per i API dettagli, consulta Command Reference. [ModifyInstancePlacement](#) AWS CLI

## modify-ipam-pool

Il seguente esempio di codice mostra come utilizzare `modify-ipam-pool`.

### AWS CLI

Per modificare un IPAM pool

L'`modify-ipam-pool` esempio seguente modifica un IPAM pool.

(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
```

Output:

```
{  
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Modificare un pool](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [ModifyIpamPool AWS CLI Command Reference](#).

## modify-ipam-resource-cidr

Il seguente esempio di codice mostra come utilizzare `modify-ipam-resource-cidr`.

### AWS CLI

Per modificare l'CIDR allocazione a una risorsa

L'`modify-ipam-resource-cidr` seguente modifica una risorsa. CIDR

(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
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni sullo spostamento delle risorse, consulta [Move resource CIDRs between scope](#) nella Amazon VPC IPAM User Guide.

Per ulteriori informazioni sulla modifica degli stati di monitoraggio, consulta [Modifica dello stato di monitoraggio della risorsa CIDRs](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [ModifyIpamResourceCidr AWS CLI Command Reference](#).

## modify-ipam-resource-discovery

Il seguente esempio di codice mostra come utilizzare `modify-ipam-resource-discovery`.

### AWS CLI

Per modificare le aree operative di una scoperta di risorse

In questo esempio, sei un amministratore IPAM delegato che desidera modificare le aree operative di una scoperta di risorse.

Per completare questa richiesta:

Non è possibile modificare un rilevamento di risorse predefinito e devi essere il proprietario del rilevamento delle risorse. È necessario l'ID di rilevamento delle risorse, che puoi utilizzare. [describe-ipam-resource-discoveries](#)

L'`modify-ipam-resource-discovery` esempio seguente modifica un rilevamento di risorse non predefinito nel tuo account. 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
```

Output:

```
{
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Work with resource discoveries](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [ModifyIpamResourceDiscovery AWS CLI Command Reference](#).

## modify-ipam-scope

Il seguente esempio di codice mostra come utilizzare `modify-ipam-scope`.

### AWS CLI

Per modificare la descrizione di un ambito

In questo scenario, sei un amministratore IPAM delegato che desidera modificare la descrizione di un IPAM ambito.

Per completare questa richiesta, avrai bisogno dell'ID dell'ambito, che puoi utilizzare. [describe-ipam-scopes](#)

L'`modify-ipam-scope` esempio seguente aggiorna la descrizione dell'ambito.

```
aws ec2 modify-ipam-scope \  
  --ipam-scope-id ipam-scope-0d3539a30b57dcdd1 \  
  --description example \  
  --region us-east-1
```

Output:

```
{
```

```
"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"
}
```

Per ulteriori informazioni sugli ambiti, consulta [How IPAM works](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [ModifyIpamScope AWS CLI Command Reference](#).

## modify-ipam

Il seguente esempio di codice mostra come utilizzare `modify-ipam`.

### AWS CLI

Per modificare un IPAM

L'esempio seguente modifica un IPAM aggiungendo una regione operativa.

(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
```

**Output:**

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [ModifyIpam](#) in AWS CLI Command Reference.

**modify-launch-template**

Il seguente esempio di codice mostra come utilizzare `modify-launch-template`.

**AWS CLI**

Per modificare la versione del modello di lancio predefinito

Questo esempio specifica la versione 2 del modello di lancio specificato come versione predefinita.

Comando:

```
aws ec2 modify-launch-template --launch-template-id lt-0abcd290751193123 --default-version 2
```

Output:

```
{
  "LaunchTemplate": {
    "LatestVersionNumber": 2,
    "LaunchTemplateId": "lt-0abcd290751193123",
    "LaunchTemplateName": "WebServers",
    "DefaultVersionNumber": 2,
    "CreatedBy": "arn:aws:iam::123456789012:root",
    "CreateTime": "2017-12-01T13:35:46.000Z"
  }
}
```

- Per API i dettagli, vedere [ModifyLaunchTemplate](#) in AWS CLI Command Reference.

## modify-managed-prefix-list

Il seguente esempio di codice mostra come utilizzare `modify-managed-prefix-list`.

### AWS CLI

Per modificare un elenco di prefissi

L'esempio seguente aggiunge una voce all'elenco di prefissi specificato.

```
aws ec2 modify-managed-prefix-list \
  --prefix-list-id pl-0123456abcabcabc1 \
  --add-entries Cidr=10.1.0.0/16,Description=vpc-c \
  --current-version 1
```

Output:

```
{
  "PrefixList": {
    "PrefixListId": "pl-0123456abcabcabc1",
    "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"
  }
}
```

Per ulteriori informazioni, consulta [Managed Prefix Lists](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [ModifyManagedPrefixList AWS CLI Command Reference](#).

## modify-network-interface-attribute

Il seguente esempio di codice mostra come utilizzare `modify-network-interface-attribute`.

### AWS CLI

Per modificare l'attributo di allegato di un'interfaccia di rete

Questo comando di esempio modifica l'`attachment` attributo dell'interfaccia di rete specificata.

Comando:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
attachment AttachmentId=eni-attach-43348162,DeleteOnTermination=false
```

Per modificare l'attributo di descrizione di un'interfaccia di rete

Questo comando di esempio modifica l'`description` attributo dell'interfaccia di rete specificata.

Comando:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
description "My description"
```

Per modificare l' `groupSet` attributo di un'interfaccia di rete

Questo comando di esempio modifica l'`groupSet` attributo dell'interfaccia di rete specificata.

Comando:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --groups sg-903004f8 sg-1a2b3c4d
```

Per modificare l' `sourceDestCheck` attributo di un'interfaccia di rete

Questo comando di esempio modifica l'`sourceDestCheck` attributo dell'interfaccia di rete specificata.

Comando:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --no-source-dest-check
```

- Per API i dettagli, vedere [ModifyNetworkInterfaceAttribute](#) in AWS CLI Command Reference.

## **modify-private-dns-name-options**

Il seguente esempio di codice mostra come utilizzare `modify-private-dns-name-options`.

AWS CLI

Per modificare le opzioni, ad esempio i nomi di host

L'`modify-private-dns-name-options` esempio seguente disabilita l'opzione di risposta alle DNS interrogazioni, ad esempio i nomi di host con record A. DNS

```
aws ec2 modify-private-dns-name-options \  
  --instance-id i-1234567890abcdef0 \  
  --no-enable-resource-name-dns-a-record
```

Output:

```
{  
  "Return": true  
}
```

Per ulteriori informazioni, consulta i [tipi di hostname delle EC2 istanze Amazon](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [ModifyPrivateDnsNameOptions AWS CLI Command Reference](#).

## modify-reserved-instances

Il seguente esempio di codice mostra come utilizzare `modify-reserved-instances`.

### AWS CLI

Per modificare le istanze riservate

Questo comando di esempio sposta un'istanza riservata in un'altra zona di disponibilità nella stessa regione.

Comando:

```
aws ec2 modify-reserved-instances --reserved-instances-ids b847fa93-e282-4f55-b59a-1342f5bd7c02 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-Classical,InstanceCount=10
```

Output:

```
{
  "ReservedInstancesModificationId": "rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687"
}
```

Per modificare la piattaforma di rete delle istanze riservate

Questo comando di esempio converte EC2 -Classic Reserved Instances in -. EC2 VPC

Comando:

```
aws ec2 modify-reserved-instances --reserved-instances-ids f127bd27-edb7-44c9-a0eb-0d7e09259af0 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-VPC,InstanceCount=5
```

Output:

```
{
  "ReservedInstancesModificationId": "rimod-82fa9020-668f-4fb6-945d-61537009d291"
}
```

Per ulteriori informazioni, consulta *Modifying Your Reserved Instances* nella *Amazon EC2 User Guide*.

Per modificare la dimensione dell'istanza delle istanze riservate

Questo comando di esempio modifica un'istanza riservata che ha 10 istanze Linux/ m1.small in us-west-1c in modo che 8 UNIX istanze m1.small diventino 2 istanze m1.large e le restanti 2 istanze m1.small diventino 1 istanza m1.medium nella stessa zona di disponibilità. Comando:

```
aws ec2 modify-reserved-instances --reserved-instances-ids 1ba8e2e3-3556-4264-949e-63ee671405a9 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-Classical,InstanceCount=2,InstanceType=m1.large AvailabilityZone=us-west-1c,Platform=EC2-Classical,InstanceCount=1,InstanceType=m1.medium
```

Output:

```
{
  "ReservedInstancesModificationId": "rimod-acc5f240-080d-4717-b3e3-1c6b11fa00b6"
}
```

Per ulteriori informazioni, consulta *Modifica della dimensione dell'istanza delle prenotazioni* nella *Amazon EC2 User Guide*.

- Per API i dettagli, consulta [ModifyReservedInstances AWS CLI Command Reference](#).

## modify-security-group-rules

Il seguente esempio di codice mostra come utilizzare `modify-security-group-rules`.

### AWS CLI

Modificare le regole di un gruppo di sicurezza per aggiornare la descrizione della regola, il protocollo IP e l'intervallo di CidrIpv4 indirizzi

L'`modify-security-group-rules` esempio seguente aggiorna la descrizione, il protocollo IP e l'IPV4CIDRintervallo di una regola del gruppo di sicurezza specificato. Utilizzate il `security-group-rules` parametro per inserire gli aggiornamenti per le regole del gruppo di sicurezza specificato. `-l` specifica tutti i protocolli.

```
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}'
```

Output:

```
{  
  "Return": true  
}
```

Per ulteriori informazioni sulle regole dei gruppi di sicurezza, consulta le [regole dei gruppi di sicurezza](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [ModifySecurityGroupRules AWS CLI Command Reference](#).

## modify-snapshot-attribute

Il seguente esempio di codice mostra come utilizzare `modify-snapshot-attribute`.

### AWS CLI

Esempio 1: modificare un attributo snapshot

L'`modify-snapshot-attribute` esempio seguente aggiorna l'`createVolumePermission` attributo per l'istantanea specificata, rimuovendo le autorizzazioni di volume per l'utente specificato.

```
aws ec2 modify-snapshot-attribute \  
--snapshot-id snap-1234567890abcdef0 \  
--attribute createVolumePermission \  
--operation-type remove \  
--user-ids 123456789012
```

Esempio 2: rendere pubblica un'istantanea

L'`modify-snapshot-attribute` esempio seguente rende pubblica l'istantanea specificata.

```
aws ec2 modify-snapshot-attribute \  
--snapshot-id snap-1234567890abcdef0 \  
--attribute createVolumePermission \  
--operation-type add \  

```

```
--group-names all
```

- Per API i dettagli, vedere [ModifySnapshotAttribute](#) in AWS CLI Command Reference.

## modify-snapshot-tier

Il seguente esempio di codice mostra come utilizzare `modify-snapshot-tier`.

### AWS CLI

Esempio 1: archiviare un'istantanea

L'`modify-snapshot-tier` esempio seguente archivia l'istantanea specificata.

```
aws ec2 modify-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --storage-tier archive
```

Output:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "TieringStartTime": "2021-09-15T16:44:37.574Z"  
}
```

Il parametro `TieringStartTime` response indica la data e l'ora in cui è stato avviato il processo di archiviazione, in formato UTC orario (YYYY-MM DDTHH -:MM:). SSZ

Per ulteriori informazioni sull'archiviazione degli snapshot, consulta Archive [Amazon EBS snapshots nella Amazon EC2 User Guide](#).

- Per API i dettagli, consulta Command [ModifySnapshotTier](#) Reference AWS CLI .

## modify-spot-fleet-request

Il seguente esempio di codice mostra come utilizzare `modify-spot-fleet-request`.

### AWS CLI

Per modificare una richiesta di flotta Spot

Questo comando di esempio aggiorna la capacità target della richiesta di flotta Spot specificata.

Comando:

```
aws ec2 modify-spot-fleet-request --target-capacity 20 --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Output:

```
{
  "Return": true
}
```

Questo comando di esempio riduce la capacità di destinazione della richiesta del parco veicoli Spot specificata senza che ciò comporti la chiusura di alcuna istanza Spot.

Comando:

```
aws ec2 modify-spot-fleet-request --target-capacity 10 --excess-capacity-termination-policy NoTermination --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

Output:

```
{
  "Return": true
}
```

- Per API i dettagli, consulta AWS CLI Command [ModifySpotFleetRequest](#)Reference.

## modify-subnet-attribute

Il seguente esempio di codice mostra come utilizzare `modify-subnet-attribute`.

### AWS CLI

Per modificare il comportamento di IPv4 indirizzamento pubblico di una sottorete

Questo esempio modifica subnet-1a2b3c4d per specificare che a tutte le istanze avviate in questa sottorete viene assegnato un indirizzo pubblico. IPv4 Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --map-public-ip-on-launch
```

Per modificare il comportamento di IPv6 indirizzamento di una sottorete

Questo esempio modifica subnet-1a2b3c4d per specificare che a tutte le istanze avviate in questa sottorete viene assegnato un indirizzo compreso nell'intervallo della sottorete. IPv6

Comando:

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --assign-ipv6-address-on-creation
```

Per ulteriori informazioni, consulta IP Addressing in Your in the Virtual Private Cloud User Guide. VPC AWS

- Per API i dettagli, vedere [ModifySubnetAttribute](#) in AWS CLI Command Reference.

## modify-traffic-mirror-filter-network-services

Il seguente esempio di codice mostra come utilizzare `modify-traffic-mirror-filter-network-services`.

AWS CLI

Per aggiungere servizi di rete a un filtro Traffic Mirror

L'`modify-traffic-mirror-filter-network-services` esempio seguente aggiunge i servizi DNS di rete Amazon al filtro specificato.

```
aws ec2 modify-traffic-mirror-filter-network-services \
  --traffic-mirror-filter-id tmf-04812ff784EXAMPLE \
  --add-network-service amazon-dns
```

Output:

```
{
  "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"
    }
]
}
}
}

```

Per ulteriori informazioni, consulta [Modify Traffic Mirror Filter Network Services](#) nella AWS Traffic Mirroring Guide.

- Per API i dettagli, vedere [ModifyTrafficMirrorFilterNetworkServices](#) in AWS CLI Command Reference.

## modify-traffic-mirror-filter-rule

Il seguente esempio di codice mostra come utilizzare `modify-traffic-mirror-filter-rule`.

### AWS CLI

Per modificare una regola del filtro Traffic Mirror

L'esempio seguente modifica la descrizione della regola di filtro Traffic Mirror specificata.

```
aws ec2 modify-traffic-mirror-filter-rule \
```

```
--traffic-mirror-filter-rule-id tmfr-0ca76e0e08EXAMPLE \  
--description "TCP Rule"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Modificare le regole del filtro Traffic Mirror](#) nella AWS Traffic Mirroring Guide.

- Per API i dettagli, consulta [ModifyTrafficMirrorFilterRule AWS CLI Command Reference](#).

## modify-traffic-mirror-session

Il seguente esempio di codice mostra come utilizzare `modify-traffic-mirror-session`.

AWS CLI

Per modificare una sessione di Traffic Mirror

L'`modify-traffic-mirror-session` seguente modifica la descrizione della sessione di Traffic Mirror e il numero di pacchetti di cui eseguire il mirroring.

```
aws ec2 modify-traffic-mirror-session \  
  --description "Change packet length" \  
  --traffic-mirror-session-id tms-08a33b1214EXAMPLE \  
  --remove-fields "packet-length"
```

Output:

```
{
  "TrafficMirrorSession": {
    "TrafficMirrorSessionId": "tms-08a33b1214EXAMPLE",
    "TrafficMirrorTargetId": "tmt-07f75d8feeEXAMPLE",
    "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
    "NetworkInterfaceId": "eni-070203f901EXAMPLE",
    "OwnerId": "111122223333",
    "SessionNumber": 1,
    "VirtualNetworkId": 7159709,
    "Description": "Change packet length",
    "Tags": []
  }
}
```

Per ulteriori informazioni, consulta [Modificare la sessione di Traffic Mirroring](#) nella Traffic Mirroring Guide.

- Per API i dettagli, consulta [ModifyTrafficMirrorSession AWS CLI Command Reference](#).

## modify-transit-gateway-prefix-list-reference

Il seguente esempio di codice mostra come utilizzare `modify-transit-gateway-prefix-list-reference`.

### AWS CLI

Per modificare un riferimento a un elenco di prefissi

L'esempio seguente modifica il riferimento all'elenco dei prefissi nella tabella delle rotte specificata cambiando l'allegato a cui viene indirizzato il traffico.

```
aws ec2 modify-transit-gateway-prefix-list-reference \
  --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
  --prefix-list-id pl-11111122222222333 \
  --transit-gateway-attachment-id tgw-attach-aabbccddaabbccaab
```

Output:

```
{
  "TransitGatewayPrefixListReference": {
```

```

    "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
    "PrefixListId": "pl-1111112222222333",
    "PrefixListOwnerId": "123456789012",
    "State": "modifying",
    "Blackhole": false,
    "TransitGatewayAttachment": {
      "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
      "ResourceType": "vpc",
      "ResourceId": "vpc-112233445566aabbcc"
    }
  }
}

```

Per ulteriori informazioni, vedete i [riferimenti all'elenco dei prefissi](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [ModifyTransitGatewayPrefixListReference](#) in AWS CLI Command Reference.

## modify-transit-gateway-vpc-attachment

Il seguente esempio di codice mostra come utilizzare `modify-transit-gateway-vpc-attachment`.

### AWS CLI

Per modificare un VPC allegato del gateway di transito

L'`modify-transit-gateway-vpc-attachment` seguente aggiunge una sottorete all'VPC allegato del gateway di transito specificato.

```

aws ec2 modify-transit-gateway-vpc-attachment \
  --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE \
  --add-subnet-ids subnet-0e51f45802EXAMPLE

```

Output:

```

{
  "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"
    }
  }
}

```

Per ulteriori informazioni, vedete [Transit gateway attachments to a VPC](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [ModifyTransitGatewayVpcAttachment](#) in AWS CLI Command Reference.

## modify-transit-gateway

Il seguente esempio di codice mostra come utilizzare `modify-transit-gateway`.

### AWS CLI

Per modificare un gateway di transito

L'esempio seguente modifica il gateway di transito specificato abilitando il ECMP supporto per VPN gli allegati.

```

aws ec2 modify-transit-gateway \
  --transit-gateway-id tgw-1111122222aaaaa \
  --options VpnEcmpSupport=enable

```

Output:

```

{
  "TransitGateway": {
    "TransitGatewayId": "tgw-1111122222aaaaa",
    "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"
    }
  }
}

```

Per ulteriori informazioni, consulta [Transit Gateways nella Transit Gateways Guide](#).

- Per API i dettagli, vedere [ModifyTransitGateway](#) in AWS CLI Command Reference.

## modify-verified-access-endpoint-policy

Il seguente esempio di codice mostra come utilizzare `modify-verified-access-endpoint-policy`.

### AWS CLI

Per configurare la politica di accesso verificato per un endpoint

L'`modify-verified-access-endpoint-policy` esempio seguente aggiunge la politica di accesso verificato specificata all'endpoint di accesso verificato specificato.

```

aws ec2 modify-verified-access-endpoint-policy \
  --verified-access-endpoint-id vae-066fac616d4d546f2 \
  --policy-enabled \
  --policy-document file://policy.txt

```

Contenuto di `policy.txt`.

```

permit(principal,action,resource)
when {
  context.identity.groups.contains("finance") &&
  context.identity.email.verified == true
};

```

**Output:**

```
{
  "PolicyEnabled": true,
  "PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
}
```

Per ulteriori informazioni, consulta le [politiche di accesso verificato](#) nella Guida per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [ModifyVerifiedAccessEndpointPolicy AWS CLI Command Reference](#).

**modify-verified-access-endpoint**

Il seguente esempio di codice mostra come utilizzare `modify-verified-access-endpoint`.

**AWS CLI**

Per modificare la configurazione di un endpoint di accesso verificato

L'`modify-verified-access-endpoint` esempio seguente aggiunge la descrizione specificata all'endpoint di accesso verificato specificato.

```
aws ec2 modify-verified-access-endpoint \
  --verified-access-endpoint-id vae-066fac616d4d546f2 \
  --description "Testing Verified Access"
```

**Output:**

```
{
  "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"
}
}

```

Per ulteriori informazioni, consulta gli [endpoint di accesso verificato nella Guida](#) per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [ModifyVerifiedAccessEndpoint AWS CLI Command Reference](#).

## modify-verified-access-group-policy

Il seguente esempio di codice mostra come utilizzare `modify-verified-access-group-policy`.

### AWS CLI

Per configurare una politica di accesso verificato per un gruppo

L'`modify-verified-access-group-policy` seguente aggiunge la politica di accesso verificato specificata al gruppo di accesso verificato specificato.

```

aws ec2 modify-verified-access-group-policy \
  --verified-access-group-id vagr-0dbe967baf14b7235 \
  --policy-enabled \
  --policy-document file://policy.txt

```

Contenuto di `policy.txt`.



```
permit(principal,action,resource)
when {
    context.identity.groups.contains("finance") &&
    context.identity.email.verified == true
};
```

Output:

```
{
  "PolicyEnabled": true,
  "PolicyDocument": "permit(principal,action,resource)\nwhen
{\n  context.identity.groups.contains(\"finance\") &&\n
context.identity.email_verified == true\n};"
}
```

Per ulteriori informazioni, consulta [Gruppi di accesso verificato](#) nella Guida per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [ModifyVerifiedAccessGroupPolicy AWS CLI Command Reference](#).

## modify-verified-access-group

Il seguente esempio di codice mostra come utilizzare `modify-verified-access-group`.

### AWS CLI

Per modificare la configurazione di un gruppo di accesso verificato

L'`modify-verified-access-group`esempio seguente aggiunge la descrizione specificata al gruppo di accesso verificato specificato.

```
aws ec2 modify-verified-access-group \
  --verified-access-group-id vagr-0dbe967baf14b7235 \
  --description "Testing Verified Access"
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Gruppi di accesso verificato](#) nella Guida per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [ModifyVerifiedAccessGroup AWS CLI Command Reference](#).

## modify-verified-access-instance-logging-configuration

Il seguente esempio di codice mostra come utilizzare `modify-verified-access-instance-logging-configuration`.

### AWS CLI

Per abilitare la registrazione per un'istanza di accesso verificato

L'`modify-verified-access-instance-logging-configuration` esempio seguente abilita la registrazione degli accessi per l'istanza di accesso verificato specificata. I log verranno consegnati al gruppo di CloudWatch log Logs specificato.

```

aws ec2 modify-verified-access-instance-logging-configuration \
  --verified-access-instance-id vai-0ce000c0b7643abea \
  --access-logs CloudWatchLogs={Enabled=true,LogGroup=my-log-group}

```

Output:

```

{
  "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
}
}
}

```

Per ulteriori informazioni, consulta i [registri di accesso verificato](#) nella Guida per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [ModifyVerifiedAccessInstanceLoggingConfiguration AWS CLI Command Reference](#).

## modify-verified-access-instance

Il seguente esempio di codice mostra come utilizzare `modify-verified-access-instance`.

### AWS CLI

Per modificare la configurazione di un'istanza di accesso verificato

L'`modify-verified-access-instance` seguente aggiunge la descrizione specificata all'istanza di accesso verificato specificata.

```

aws ec2 modify-verified-access-instance \
  --verified-access-instance-id vai-0ce000c0b7643abea \
  --description "Testing Verified Access"

```

Output:

```

{
  "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"
}
```

Per ulteriori informazioni, consulta le [istanze di accesso verificato](#) nella Guida per l'utente di accesso AWS verificato.

- Per API i dettagli, consulta [ModifyVerifiedAccessInstance AWS CLI Command Reference](#).

## modify-verified-access-trust-provider

Il seguente esempio di codice mostra come utilizzare `modify-verified-access-trust-provider`.

### AWS CLI

Per modificare la configurazione di un provider fiduciario ad accesso verificato

L'`modify-verified-access-trust-provider` seguente aggiunge la descrizione specificata al provider fiduciario Verified Access specificato.

```
aws ec2 modify-verified-access-trust-provider \
  --verified-access-trust-provider-id vatp-0bb32de759a3e19e7 \
  --description "Testing Verified Access"
```

Output:

```
{
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Trust providers for Verified Access](#) nella AWS Verified Access User Guide.

- Per API i dettagli, consulta [ModifyVerifiedAccessTrustProvider AWS CLI Command Reference](#).

## modify-volume-attribute

Il seguente esempio di codice mostra come utilizzare `modify-volume-attribute`.

### AWS CLI

Per modificare un attributo di volume

Questo esempio imposta l'attributo `autoEnableIo` del volume con ID `vol-1234567890abcdef0` su `true`. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 modify-volume-attribute --volume-id vol-1234567890abcdef0 --auto-enable-io
```

- Per API i dettagli, vedere [ModifyVolumeAttribute](#) in AWS CLI Command Reference.

## modify-volume

Il seguente esempio di codice mostra come utilizzare `modify-volume`.

### AWS CLI

Esempio 1: modificare un volume cambiandone le dimensioni

L'esempio seguente modifica la dimensione del volume specificato a 150 GB.

Comando:

```
aws ec2 modify-volume --size 150 --volume-id vol-1234567890abcdef0
```

## Output:

```
{
  "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
  }
}
```

Esempio 2: modificare un volume modificandone il tipo, la dimensione e IOPS il valore

L'`modify-volume` seguente modifica il tipo di volume in Provisioned IOPS SSD, imposta la IOPS frequenza target su 10000 e imposta la dimensione del volume su 350 GB.

```
aws ec2 modify-volume \
  --volume-type io1 \
  --iops 10000 \
  --size 350 \
  --volume-id vol-1234567890abcdef0
```

## Output:

```
{
  "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
  }
}
```

```
}  
}
```

- Per API i dettagli, vedere [ModifyVolume](#) in AWS CLI Command Reference.

## modify-vpc-attribute

Il seguente esempio di codice mostra come utilizzare `modify-vpc-attribute`.

### AWS CLI

Per modificare l' `enableDnsSupport` attributo

Questo esempio modifica l'`enableDnsSupport` attributo. Questo attributo indica se DNS la risoluzione è abilitata per VPC. Se questo attributo è `true`, il DNS server Amazon risolve i DNS nomi host delle tue istanze negli indirizzi IP corrispondenti; in caso contrario, non lo fa. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-support "{\"Value  
\":false}"
```

Per modificare l'attributo `enableDnsHostnames`

Questo esempio modifica l'`enableDnsHostnames` attributo. Questo attributo indica se le istanze sono state avviate nel comando VPC get DNS hostnames. Se questo attributo è `true`, le istanze contenute in VPC get DNS hostnames; in caso contrario, non lo fanno. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-hostnames "{\"Value  
\":false}"
```

- Per API i dettagli, vedere [ModifyVpcAttribute](#) in AWS CLI Command Reference.

## modify-vpc-endpoint-connection-notification

Il seguente esempio di codice mostra come utilizzare `modify-vpc-endpoint-connection-notification`.

## AWS CLI

Per modificare una notifica di connessione all'endpoint

Questo esempio modifica l'SNSargomento della notifica di connessione all'endpoint specificata.

Comando:

```
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
```

Output:

```
{  
  "ReturnValue": true  
}
```

- Per API i dettagli, vedere [ModifyVpcEndpointConnectionNotification](#) in AWS CLI Command Reference.

## modify-vpc-endpoint-service-configuration

Il seguente esempio di codice mostra come utilizzare `modify-vpc-endpoint-service-configuration`.

## AWS CLI

Per modificare la configurazione di un servizio endpoint

Questo esempio modifica il requisito di accettazione per il servizio endpoint specificato.

Comando:

```
aws ec2 modify-vpc-endpoint-service-configuration --service-id vpce-svc-09222513e6e77dc86 --no-acceptance-required
```

Output:

```
{
```



```
"ReturnValue": true
}
```

- Per API i dettagli, vedere [ModifyVpcEndpointServiceConfiguration](#) in AWS CLI Command Reference.

## **modify-vpc-endpoint-service-payer-responsibility**

Il seguente esempio di codice mostra come utilizzare `modify-vpc-endpoint-service-payer-responsibility`.

### AWS CLI

Per modificare la responsabilità del pagatore

L'esempio seguente modifica la responsabilità del pagatore del servizio endpoint specificato.

```
aws ec2 modify-vpc-endpoint-service-payer-responsibility \
  --service-id vpce-svc-071afff70666e61e0 \
  --payer-responsibility ServiceOwner
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [ModifyVpcEndpointServicePayerResponsibility](#) in AWS CLI Command Reference.

## **modify-vpc-endpoint-service-permissions**

Il seguente esempio di codice mostra come utilizzare `modify-vpc-endpoint-service-permissions`.

### AWS CLI

Per modificare le autorizzazioni del servizio endpoint

Questo esempio aggiunge l'autorizzazione per un AWS account a connettersi al servizio endpoint specificato.

Comando:

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-  
svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:root"]'
```

Output:

```
{  
  "ReturnValue": true  
}
```

Questo esempio aggiunge l'autorizzazione per un IAM utente specifico (admin) a connettersi al servizio endpoint specificato.

Comando:

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-  
svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:user/  
admin"]'
```

- Per API i dettagli, vedere [ModifyVpcEndpointServicePermissions](#) in AWS CLI Command Reference.

## modify-vpc-endpoint

Il seguente esempio di codice mostra come utilizzare `modify-vpc-endpoint`.

AWS CLI

Per modificare un endpoint del gateway

Questo esempio modifica l'endpoint del gateway `vpce-1a2b3c4d` associando la tabella di routing all'endpoint e `rtb-aaa222bb` reimpostando il documento di policy.

Comando:

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-1a2b3c4d --add-route-table-  
ids rtb-aaa222bb --reset-policy
```

Output:

```
{
```

```
"Return": true
}
```

Per modificare un endpoint di interfaccia

Questo esempio modifica l'endpoint dell'interfaccia `vpce-0fe5b17a0707d6fa5` aggiungendo una sottorete `subnet-d6fcaa8d` all'endpoint.

Comando:

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-0fe5b17a0707d6fa5 --add-subnet-id subnet-d6fcaa8d
```

Output:

```
{
  "Return": true
}
```

- Per i API dettagli, vedere [ModifyVpcEndpoint](#) in Command Reference.AWS CLI

## modify-vpc-peering-connection-options

Il seguente esempio di codice mostra come utilizzare `modify-vpc-peering-connection-options`.

AWS CLI

Per abilitare la comunicazione tramite una connessione VPC peering dalla connessione locale ClassicLink

In questo esempio, per la connessione peering `pcx-aaaabbbb`, il proprietario del richiedente VPC modifica le opzioni di connessione VPC peering per consentire a una ClassicLink connessione locale di comunicare con il peer. VPC

Comando:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-aaaabbbb --requester-peering-connection-options AllowEgressFromLocalClassicLinkToRemoteVpc=true
```

## Output:

```
{
  "RequesterPeeringConnectionOptions": {
    "AllowEgressFromLocalClassicLinkToRemoteVpc": true
  }
}
```

Per abilitare la comunicazione tramite una connessione VPC peering dalla connessione locale a una remota VPC ClassicLink

In questo esempio, il proprietario dell'accettatore VPC modifica le opzioni di connessione VPC peering per consentire alla connessione locale VPC di comunicare con la ClassicLink connessione nel peer. VPC

## Comando:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-aaaabbbb --accepter-peering-connection-options AllowEgressFromLocalVpcToRemoteClassicLink=true
```

## Output:

```
{
  "AcceptorPeeringConnectionOptions": {
    "AllowEgressFromLocalVpcToRemoteClassicLink": true
  }
}
```

Per abilitare il supporto alla DNS risoluzione per la connessione peering VPC

In questo esempio, il proprietario del richiedente VPC modifica le opzioni di connessione VPC peering per consentire pcx-aaaabbbb al locale VPC di risolvere i DNS nomi di host pubblici in indirizzi IP privati quando vengono richieste richieste dalle istanze del peer. VPC

## Comando:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-aaaabbbb --requester-peering-connection-options AllowDnsResolutionFromRemoteVpc=true
```

## Output:

```
{
  "RequesterPeeringConnectionOptions": {
    "AllowDnsResolutionFromRemoteVpc": true
  }
}
```

- Per i dettagli, vedere in Command Reference. API [ModifyVpcPeeringConnectionOptions](#) AWS CLI

## modify-vpc-tenancy

Il seguente esempio di codice mostra come utilizzare `modify-vpc-tenancy`.

AWS CLI

Per modificare la locazione di un VPC

Questo esempio modifica la locazione di to. VPC `vpc-1a2b3c4d` `default`

Comando:

```
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```

Output:

```
{
  "Return": true
}
```

- Per API i dettagli, vedere [ModifyVpcTenancy](#) in AWS CLI Command Reference.

## modify-vpn-connection-options

Il seguente esempio di codice mostra come utilizzare `modify-vpn-connection-options`.

AWS CLI

Per modificare le opzioni di VPN connessione

L'`modify-vpn-connection-options` esempio seguente modifica la connessione locale IPv4 CIDR sul lato gateway del cliente della VPN connessione specificata.

```
aws ec2 modify-vpn-connection-options \  
--vpn-connection-id vpn-1122334455aabbccd \  
--local-ipv4-network-cidr 10.0.0.0/16
```

Output:

```
{  
  "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": ""
  }
]
}

```

Per ulteriori informazioni, vedere [Modifica delle opzioni di Site-to-Site VPN connessione nella Guida per l'utente per la versione 1](#).

- Per API i dettagli, vedere [ModifyVpnConnectionOptions](#) in AWS CLI Command Reference.

## modify-vpn-connection

Il seguente esempio di codice mostra come utilizzare `modify-vpn-connection`.

### AWS CLI

Per modificare una VPN connessione

L'esempio seguente modifica il gateway di destinazione per VPN la connessione `vpn-12345678901234567` al gateway privato virtuale `vgw-11223344556677889`:

```

aws ec2 modify-vpn-connection \
  --vpn-connection-id vpn-12345678901234567 \
  --vpn-gateway-id vgw-11223344556677889

```

Output:

```

{
  "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"
    }
  ]
}

```

- Per API i dettagli, vedere [ModifyVpnConnection](#) in AWS CLI Command Reference.

## modify-vpn-tunnel-certificate

Il seguente esempio di codice mostra come utilizzare `modify-vpn-tunnel-certificate`.

### AWS CLI

Per ruotare un certificato di VPN tunnel

L'`modify-vpn-tunnel-certificate` esempio seguente ruota il certificato per il tunnel specificato per una connessione VPN

```

aws ec2 modify-vpn-tunnel-certificate \
  --vpn-tunnel-outside-ip-address 203.0.113.17 \
  --vpn-connection-id vpn-12345678901234567

```

Output:

```

{
  "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"
      }
    ]
  }
}

```

- Per API i dettagli, vedere [ModifyVpnTunnelCertificate](#) in AWS CLI Command Reference.

## modify-vpn-tunnel-options

Il seguente esempio di codice mostra come utilizzare `modify-vpn-tunnel-options`.

### AWS CLI

Per modificare le opzioni del tunnel per una VPN connessione

L'`modify-vpn-tunnel-options` esempio seguente aggiorna i gruppi Diffie-Hellman consentiti per il tunnel e la connessione specificati. 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}]
```

Output:

```
{  
  "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"
  }
]
}
}
```

- Per i API dettagli, vedere [ModifyVpnTunnelOptions](#) in Command Reference.AWS CLI

## monitor-instances

Il seguente esempio di codice mostra come utilizzare `monitor-instances`.

### AWS CLI

Per abilitare il monitoraggio dettagliato per un'istanza

Questo comando di esempio abilita il monitoraggio dettagliato per l'istanza specificata.

Comando:

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

Output:

```
{
  "InstanceMonitorings": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "Monitoring": {
        "State": "pending"
      }
    }
  ]
}
```

- Per API i dettagli, vedere [MonitorInstances](#) in AWS CLI Command Reference.

## move-address-to-vpc

Il seguente esempio di codice mostra come utilizzare `move-address-to-vpc`.

### AWS CLI

Per spostare un indirizzo in EC2 - VPC

Questo esempio sposta l'indirizzo IP elastico 54.123.4.56 sulla piattaforma -. EC2 VPC

Comando:

```
aws ec2 move-address-to-vpc --public-ip 54.123.4.56
```

Output:

```
{
  "Status": "MoveInProgress"
}
```

- Per i API dettagli, vedere [MoveAddressToVpc](#) in Command Reference.AWS CLI

## move-byoip-cidr-to-ipam

Il seguente esempio di codice mostra come utilizzare `move-byoip-cidr-to-ipam`.

AWS CLI

Per trasferire BYOIP CIDR un IPAM

L'`move-byoip-cidr-to-ipam` seguente trasferisce un BYOIP CIDR a 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
```

Output:

```
{
  "ByoipCidr": {
    "Cidr": "130.137.249.0/24",
    "State": "pending-transfer"
  }
}
```

Per ulteriori informazioni, consulta [Tutorial: Trasferisci un BYOIP IPv4 CIDR indirizzo esistente IPAM](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [MoveByoipCidrToIpam AWS CLI Command Reference](#).

## network-insights-access-scope

Il seguente esempio di codice mostra come utilizzare `network-insights-access-scope`.

### AWS CLI

Per creare ambiti di accesso a Network Insights

L'esempio seguente crea un ambito di accesso a Network Insights nel tuo AWS account.

```
aws ec2 create-network-insights-access-scope \  
  --cli-input-json file://access-scope-file.json
```

Contenuto di `access-scope-file.json`.

```
{  
  {  
    "MatchPaths": [  
      {  
        "Source": {  
          "ResourceStatement": {  
            "Resources": [  
              "vpc-abcd12e3"  
            ]  
          }  
        }  
      ],  
      "ExcludePaths": [  
        {  
          "Source": {  
            "ResourceStatement": {  
              "ResourceTypes": [  
                "AWS::EC2::InternetGateway"  
              ]  
            }  
          }  
        }  
      ]  
    }  
  }  
}
```

```

    }
  ]
}

```

### Output:

```

{
  "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"
            ]
          }
        }
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Guida introduttiva a Network Access Analyzer utilizzando la AWS CLI](#) guida Network Access Analyzer.

- Per API i dettagli, vedere [NetworkInsightsAccessScope](#) in AWS CLI Command Reference.

## provision-byoip-cidr

Il seguente esempio di codice mostra come utilizzare `provision-byoip-cidr`.

### AWS CLI

Per fornire un intervallo di indirizzi

L'esempio seguente fornisce un intervallo di indirizzi IP pubblico da utilizzare con AWS.

```
aws ec2 provision-byoip-cidr \
  --cidr 203.0.113.25/24 \
  --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

Output:

```
{
  "ByoipCidr": {
    "Cidr": "203.0.113.25/24",
    "State": "pending-provision"
  }
}
```

Per ulteriori informazioni sulla creazione delle stringhe dei messaggi per il contesto di autorizzazione, consulta [Bring Your Own IP Addresses](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [ProvisionByoipCidr AWS CLI](#) Command Reference.

## provision-ipam-pool-cidr

Il seguente esempio di codice mostra come utilizzare `provision-ipam-pool-cidr`.

### AWS CLI

Fornire un file CIDR a un IPAM pool



L'`provision-ipam-pool-cidr` seguente fornisce CIDR a un IPAM pool.

(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
```

Output:

```
{  
  "IpamPoolCidr": {  
    "Cidr": "10.0.0.0/24",  
    "State": "pending-provision"  
  }  
}
```

Per ulteriori informazioni, consulta la sezione [Fornitura CIDRs a un pool](#) nella Amazon VPC IPAM User Guide.

- Per API i dettagli, consulta [ProvisionIpamPoolCidr AWS CLI Command Reference](#).

## **purchase-host-reservation**

Il seguente esempio di codice mostra come utilizzare `purchase-host-reservation`.

AWS CLI

Per acquistare una prenotazione dedicata agli host

Questo esempio acquista l'offerta di prenotazione dell'host dedicato specificata per l'host dedicato specificato nel tuo account.

Comando:

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --host-id-set h-013abcd2a00cbd123
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, consulta [PurchaseHostReservation AWS CLI Command Reference](#).

## purchase-reserved-instances-offering

Il seguente esempio di codice mostra come utilizzare `purchase-reserved-instances-offering`.

### AWS CLI

Per acquistare un'offerta di istanze riservate

Questo comando di esempio illustra l'acquisto di un'offerta di istanze riservate, specificando l'ID dell'offerta e il numero di istanze.

Comando:

```
aws ec2 purchase-reserved-instances-offering --reserved-instances-offering-id ec06327e-dd07-46ee-9398-75b5fexample --instance-count 3
```

Output:

```
{
  "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"
}
```

- Per API i dettagli, vedere [PurchaseReservedInstancesOffering](#) in AWS CLI Command Reference.

## **purchase-scheduled-instances**

Il seguente esempio di codice mostra come utilizzare `purchase-scheduled-instances`.

AWS CLI

Per acquistare un'istanza pianificata

Questo esempio acquista un'istanza pianificata.

Comando:

```
aws ec2 purchase-scheduled-instances --purchase-requests file://purchase-request.json
```

Purchase-request.json:

```
[
  {
    "PurchaseToken": "eyJ2IjoiMSIsInMiOjEsImMiOi...",
    "InstanceCount": 1
  }
]
```

Output:

```
{
  "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"
  }
]
}
```

- Per i API dettagli, vedere in Command Reference. [PurchaseScheduledInstances](#) AWS CLI

## reboot-instances

Il seguente esempio di codice mostra come utilizzare `reboot-instances`.

### AWS CLI

Per riavviare un'istanza Amazon EC2

Questo esempio riavvia l'istanza specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 reboot-instances --instance-ids i-1234567890abcdef5
```

Per ulteriori informazioni, consulta Riavvio dell'istanza nella Guida per l'utente di Amazon Elastic Compute Cloud.

- Per API i dettagli, consulta [RebootInstances AWS CLI](#) Command Reference.

## register-image

Il seguente esempio di codice mostra come utilizzare `register-image`.

### AWS CLI

Esempio 1: registrare e AMI utilizzare un file manifest

L'`register-image`esempio seguente registra un AMI utilizzo del file manifest specificato in Amazon S3.

```
aws ec2 register-image \  
  --name my-image \  
  --image-location my-s3-bucket/myimage/image.manifest.xml
```

Output:

```
{  
  "ImageId": "ami-1234567890EXAMPLE"  
}
```

Per ulteriori informazioni, consulta [Amazon Machine Images \(AMI\)](#) nella Amazon EC2 User Guide.

Esempio 2: registrare e AMI utilizzare un'istantanea di un dispositivo root

L'`register-image`esempio seguente registra un AMI utilizzo dell'istantanea specificata di un volume EBS root come dispositivo. `/dev/xvda` La mappatura dei dispositivi a blocchi include anche un volume vuoto da 100 EBS GiB come dispositivo. `/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}
```

Output:

```
{  
  "ImageId": "ami-1a2b3c4d5eEXAMPLE"  
}
```

```
}
```

Per ulteriori informazioni, consulta [Amazon Machine Images \(AMI\)](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [RegisterImage AWS CLI Command Reference](#).

## register-instance-event-notification-attributes

Il seguente esempio di codice mostra come utilizzare `register-instance-event-notification-attributes`.

### AWS CLI

Esempio 1: includere tutti i tag nelle notifiche degli eventi

L'`register-instance-event-notification-attributes` esempio seguente include tutti i tag nelle notifiche degli eventi.

```
aws ec2 register-instance-event-notification-attributes \  
  --instance-tag-attribute IncludeAllTagsOfInstance=true
```

Output:

```
{  
  "InstanceTagAttribute": {  
    "InstanceTagKeys": [],  
    "IncludeAllTagsOfInstance": true  
  }  
}
```

Per ulteriori informazioni, consulta [Eventi pianificati per le tue istanze](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

Esempio 2: per includere tag specifici nelle notifiche degli eventi

L'`register-instance-event-notification-attributes` esempio seguente include i tag specificati nelle notifiche degli eventi. Se lo è, non `IncludeAllTagsOfInstance` è possibile specificare i tag `true`.

```
aws ec2 register-instance-event-notification-attributes \  
  --instance-tag-attribute InstanceTagKeys="tag-key1","tag-key2"
```

**Output:**

```
{
  "InstanceTagAttribute": {
    "InstanceTagKeys": [
      "tag-key1",
      "tag-key2"
    ],
    "IncludeAllTagsOfInstance": false
  }
}
```

Per ulteriori informazioni, consulta [Eventi pianificati per le tue istanze](#) nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

- Per API i dettagli, consulta Command [RegisterInstanceEventNotificationAttributesReference](#) AWS CLI .

**register-transit-gateway-multicast-group-sources**

Il seguente esempio di codice mostra come utilizzare `register-transit-gateway-multicast-group-sources`.

**AWS CLI**

Per registrare una sorgente con un gruppo multicast del gateway di transito.

L'`register-transit-gateway-multicast-group-sources`esempio seguente registra l'origine del gruppo di interfacce di rete specificato con un gruppo multicast.

```
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
```

**Output:**

```
{
  "RegisteredMulticastGroupSources": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
    "RegisteredNetworkInterfaceIds": [
      "eni-07f290fc3c090cbae"
    ]
  }
}
```

```

    ],
    "GroupIpAddress": "224.0.1.0"
  }
}

```

Per ulteriori informazioni, vedete [Register Sources with a Multicast Group](#) nella AWS Transit Gateways User Guide.

- Per API i dettagli, vedere [RegisterTransitGatewayMulticastGroupSources](#) in AWS CLI Command Reference.

## register-transit-gateway-multicast-group-members

Il seguente esempio di codice mostra come utilizzare `register-transit-gateway-multicast-group-members`.

### AWS CLI

Per visualizzare le informazioni sulle associazioni di domini multicast del gateway di transito

L'`register-transit-gateway-multicast-group-members` esempio seguente restituisce le associazioni per il dominio multicast specificato.

```

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

```

Output:

```

{
  "RegisteredMulticastGroupMembers": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
    "RegisteredNetworkInterfaceIds": [
      "eni-0e246d32695012e81"
    ],
    "GroupIpAddress": "224.0.1.0"
  }
}

```

Per ulteriori informazioni, vedere [Managing multicast domain](#) nella Transit Gateways User Guide.



- Per API i dettagli, vedere [RegisterTransitGatewayMulticastGroupMembers](#) in AWS CLI Command Reference.

## register-transit-gateway-multicast-group-sources

Il seguente esempio di codice mostra come utilizzare `register-transit-gateway-multicast-group-sources`.

### AWS CLI

Per registrare una sorgente con un gruppo multicast del gateway di transito.

L'esempio seguente registra l'origine del gruppo di interfacce di rete specificato con un gruppo multicast.

```
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
```

Output:

```
{
  "RegisteredMulticastGroupSources": {
    "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
    "RegisteredNetworkInterfaceIds": [
      "eni-07f290fc3c090cbae"
    ],
    "GroupIpAddress": "224.0.1.0"
  }
}
```

Per ulteriori informazioni, vedere [Managing multicast Domains](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [RegisterTransitGatewayMulticastGroupSources](#) in AWS CLI Command Reference.

## reject-transit-gateway-peering-attachment

Il seguente esempio di codice mostra come utilizzare `reject-transit-gateway-peering-attachment`.

## AWS CLI

Per rifiutare un allegato di peering del gateway di transito

L'esempio seguente rifiuta la richiesta di allegato di peering del gateway di transito specificata. Il `--region` parametro specifica la regione in cui si trova il gateway di transito accettante.

```
aws ec2 reject-transit-gateway-peering-attachment \  
  --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \  
  --region us-east-2
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Transit Gateway Peering Attachments](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [RejectTransitGatewayPeeringAttachment](#) in AWS CLI Command Reference.

## **reject-transit-gateway-vpc-attachment**

Il seguente esempio di codice mostra come utilizzare `reject-transit-gateway-vpc-attachment`.

## AWS CLI

Per rifiutare un allegato del gateway VPC di transito

L'esempio seguente rifiuta l'allegato del gateway VPC di transito specificato.

```
aws ec2 reject-transit-gateway-vpc-attachment \  
  --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, vedete [Transit gateway attachments to a VPC](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [RejectTransitGatewayVpcAttachment](#) in AWS CLI Command Reference.

## **reject-transit-gateway-vpc-attachments**

Il seguente esempio di codice mostra come utilizzare `reject-transit-gateway-vpc-attachments`.

## AWS CLI

Per rifiutare un allegato del gateway VPC di transito

L'esempio seguente rifiuta l'allegato del gateway VPC di transito specificato.

```
aws ec2 reject-transit-gateway-vpc-attachment \  
  --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, vedete [Transit gateway attachments to a VPC](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [RejectTransitGatewayVpcAttachments](#) in AWS CLI Command Reference.

## **reject-vpc-endpoint-connections**

Il seguente esempio di codice mostra come utilizzare `reject-vpc-endpoint-connections`.

## AWS CLI

Per rifiutare una richiesta di connessione all'endpoint dell'interfaccia

Questo esempio rifiuta la richiesta di connessione all'endpoint specificata per il servizio endpoint specificato.

Comando:

```
aws ec2 reject-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --  
vpc-endpoint-ids vpce-0c1308d7312217abc
```

Output:

```
{  
  "Unsuccessful": []  
}
```

- Per API i dettagli, vedere [RejectVpcEndpointConnections](#) in AWS CLI Command Reference.

## **reject-vpc-peering-connection**

Il seguente esempio di codice mostra come utilizzare `reject-vpc-peering-connection`.

## AWS CLI

Per rifiutare una connessione VPC peering

Questo esempio rifiuta la richiesta di connessione VPC peering specificata.

Comando:

```
aws ec2 reject-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

Output:

```
{  
  "Return": true  
}
```

- Per API i dettagli, vedere [RejectVpcPeeringConnection](#) in AWS CLI Command Reference.

## release-address

Il seguente esempio di codice mostra come utilizzare `release-address`.

### AWS CLI

Per rilasciare un indirizzo IP elastico per EC2 -Classic

Questo esempio rilascia un indirizzo IP elastico da utilizzare con le istanze in EC2 -Classic. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 release-address --public-ip 198.51.100.0
```

Per rilasciare un indirizzo IP elastico per - EC2 VPC

Questo esempio rilascia un indirizzo IP elastico da utilizzare con le istanze in aVPC. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 release-address --allocation-id eipalloc-64d5890a
```

- Per API i dettagli, vedere [ReleaseAddress](#) in AWS CLI Command Reference.

## release-hosts

Il seguente esempio di codice mostra come utilizzare `release-hosts`.

### AWS CLI

Per rilasciare un host dedicato dal tuo account

Per rilasciare un host dedicato dal tuo account. Le istanze presenti sull'host devono essere interrotte o terminate prima che l'host possa essere rilasciato.

Comando:

```
aws ec2 release-hosts --host-id=h-0029d6e3cacf1b3da
```

**Output:**

```
{
  "Successful": [
    "h-0029d6e3cacf1b3da"
  ],
  "Unsuccessful": []
}
```

- Per API i dettagli, vedere [ReleaseHosts](#) in AWS CLI Command Reference.

**release-ipam-pool-allocation**

Il seguente esempio di codice mostra come utilizzare `release-ipam-pool-allocation`.

**AWS CLI**

Per rilasciare un'allocazione di IPAM pool

In questo esempio, sei un amministratore IPAM delegato che ha provato a eliminare un IPAM pool ma ha ricevuto un errore secondo cui non puoi eliminare il pool mentre il pool dispone di allocazioni. Stai utilizzando questo comando per rilasciare l'allocazione di un pool.

Tieni presente quanto segue:

È possibile utilizzare questo comando solo per allocazioni personalizzate. Per rimuovere un'allocazione per una risorsa senza eliminarla, imposta lo stato monitorato su false utilizzando [modify-ipam-resource-cidr](#). Per completare questa richiesta, avrai bisogno dell'ID del IPAM pool, che puoi utilizzare. [describe-ipam-pools](#) È inoltre necessario l'ID di allocazione, che è possibile ottenere con [get-ipam-pool-allocations](#). Se non si desidera rimuovere le allocazioni una per una, è possibile utilizzare `--cascade` option quando si elimina un IPAM pool per rilasciare automaticamente tutte le allocazioni nel pool prima di eliminarlo. Esistono diversi prerequisiti prima di eseguire questo comando. Per ulteriori informazioni, consulta [Release an allocation](#) nella Amazon VPC IPAM User Guide. Il comando `--region` in cui esegui questo comando deve essere la lingua del IPAM pool in cui si trova l'allocazione.

L'`release-ipam-pool-allocation` esempio seguente rilascia un'IPAM allocazione di pool.

```
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
```

Output:

```
{  
  "Success": true  
}
```

Dopo aver rilasciato un'allocazione, potresti voler eseguirla. [delete-ipam-pool](#)

- Per API i dettagli, vedere [ReleaseIpamPoolAllocation](#) in AWS CLI Command Reference.

## replace-iam-instance-profile-association

Il seguente esempio di codice mostra come utilizzare `replace-iam-instance-profile-association`.

AWS CLI

Per sostituire un profilo di IAM istanza con un'istanza

Questo esempio sostituisce il profilo di IAM istanza rappresentato dall'associazione `iip-assoc-060bae234aac2e7fa` con il profilo di IAM istanza denominato `AdminRole`.

```
aws ec2 replace-iam-instance-profile-association \  
--iam-instance-profile Name=AdminRole \  
--association-id iip-assoc-060bae234aac2e7fa
```

Output:

```
{  
  "IamInstanceProfileAssociation": {  
    "InstanceId": "i-087711ddaf98f9489",  
    "State": "associating",  
    "AssociationId": "iip-assoc-0b215292fab192820",  
    "IamInstanceProfile": {  
      "Id": "AIPAJLNLDX3AMYZNWYYAY",  
      "Arn": "arn:aws:iam::123456789012:instance-profile/AdminRole"  
    }  
  }  
}
```



```
}
```

- Per API i dettagli, vedere [ReplacelamInstanceProfileAssociation](#) in AWS CLI Command Reference.

## replace-network-acl-association

Il seguente esempio di codice mostra come utilizzare `replace-network-acl-association`.

### AWS CLI

Per sostituire la rete ACL associata a una sottorete

Questo esempio associa la rete ACL specificata alla sottorete per l'associazione di rete specificata. ACL

Comando:

```
aws ec2 replace-network-acl-association --association-id aiclassoc-e5b95c8c --  
network-acl-id acl-5fb85d36
```

Output:

```
{  
  "NewAssociationId": "aiclassoc-3999875b"  
}
```

- Per API i dettagli, vedere [ReplaceNetworkAclAssociation](#) in AWS CLI Command Reference.

## replace-network-acl-entry

Il seguente esempio di codice mostra come utilizzare `replace-network-acl-entry`.

### AWS CLI

Per sostituire una ACL voce di rete

Questo esempio sostituisce una voce per la rete ACL specificata. La nuova regola 100 consente il traffico in ingresso da 203.0.113.12/24 sulla UDP porta 53 () in qualsiasi sottorete associata. DNS

Comando:

```
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
```

- Per i dettagli, vedere in Command Reference. API [ReplaceNetworkAclEntry](#) AWS CLI

## replace-route-table-association

Il seguente esempio di codice mostra come utilizzare `replace-route-table-association`.

### AWS CLI

Per sostituire la tabella di routing associata a una sottorete

Questo esempio associa la tabella di routing specificata alla subnet per l'associazione della tabella di routing specificata.

Comando:

```
aws ec2 replace-route-table-association --association-id rtbassoc-781d0d1a --route-table-id rtb-22574640
```

Output:

```
{
  "NewAssociationId": "rtbassoc-3a1f0f58"
}
```

- Per API i dettagli, vedere [ReplaceRouteTableAssociation](#) in AWS CLI Command Reference.

## replace-route

Il seguente esempio di codice mostra come utilizzare `replace-route`.

### AWS CLI

Per sostituire un percorso

Questo esempio sostituisce la rotta specificata nella tabella delle rotte specificata. La nuova route corrisponde a quella specificata CIDR e invia il traffico al gateway privato virtuale specificato. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 replace-route --route-table-id rtb-22574640 --destination-cidr-block 10.0.0.0/16 --gateway-id vgw-9a4cacf3
```

- Per API i dettagli, vedere [ReplaceRoute](#) in AWS CLI Command Reference.

## replace-transit-gateway-route

Il seguente esempio di codice mostra come utilizzare `replace-transit-gateway-route`.

### AWS CLI

Per sostituire il percorso specificato nella tabella delle rotte del gateway di transito specificato

L'`replace-transit-gateway-route` esempio seguente sostituisce il percorso nella tabella delle rotte del gateway di transito specificato.

```
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-0a823edbbeEXAMPLE
```

Output:

```
{
  "Route": {
    "DestinationCidrBlock": "10.0.2.0/24",
    "TransitGatewayAttachments": [
      {
        "ResourceId": "vpc-4EXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
        "ResourceType": "vpc"
      }
    ],
    "Type": "static",
    "State": "active"
  }
}
```

Per ulteriori informazioni, consulta le [tabelle delle rotte dei gateway di transito](#) nella Guida ai gateway di transito.

- Per API i dettagli, vedere [ReplaceTransitGatewayRoute](#) in AWS CLI Command Reference.

## report-instance-status

Il seguente esempio di codice mostra come utilizzare `report-instance-status`.

### AWS CLI

Per segnalare il feedback sullo stato di un'istanza

Questo comando di esempio riporta il feedback sullo stato dell'istanza specificata.

Comando:

```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status impaired --reason-codes unresponsive
```

- Per API i dettagli, vedere [ReportInstanceStatus](#) in AWS CLI Command Reference.

## request-spot-fleet

Il seguente esempio di codice mostra come utilizzare `request-spot-fleet`.

### AWS CLI

Per richiedere una flotta Spot nella sottorete al prezzo più basso

Questo comando di esempio crea una richiesta di flotta Spot con due specifiche di lancio che differiscono solo in base alla sottorete. La flotta Spot avvia le istanze nella sottorete specificata al prezzo più basso. Se le istanze vengono avviate in modo predefinito VPC, ricevono per impostazione predefinita un indirizzo IP pubblico. Se le istanze vengono avviate in modo non predefinito VPC, per impostazione predefinita non ricevono un indirizzo IP pubblico.

Tieni presente che non puoi specificare sottoreti diverse dalla stessa zona di disponibilità in una richiesta di flotta Spot.

Comando:

```
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"
      }
    }
  ]
}
```

Output:

```
{
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

Per richiedere una flotta Spot nella zona di disponibilità al prezzo più basso

Questo comando di esempio crea una richiesta di flotta Spot con due specifiche di lancio che differiscono solo in base alla zona di disponibilità. La flotta Spot avvia le istanze nella zona di disponibilità specificata al prezzo più basso. Se il tuo account supporta EC2 VPC solo le istanze Spot, Amazon EC2 avvia le istanze Spot nella sottorete predefinita della zona di disponibilità. Se il tuo account supporta EC2 -Classic, Amazon EC2 avvia le istanze in EC2 -Classic nella zona di disponibilità.

Comando:

```
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"
      }
    }
  ]
}
```

Per avviare istanze Spot in una sottorete e assegnare loro indirizzi IP pubblici

Questo comando di esempio assegna indirizzi pubblici alle istanze avviate in modo non predefinito. VPC Si noti che quando si specifica un'interfaccia di rete, è necessario includere l'ID di sottorete e l'ID del gruppo di sicurezza utilizzando l'interfaccia di rete.

Comando:

```
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"
    }
  }
]
}
```

Per richiedere una flotta Spot utilizzando la strategia di allocazione diversificata

Questo comando di esempio crea una richiesta di flotta Spot che avvia 30 istanze utilizzando la strategia di allocazione diversificata. Le specifiche di lancio variano in base al tipo di istanza. Il parco istanze Spot distribuisce le istanze tra le specifiche di lancio, in modo che vi siano 10 istanze per ogni tipo.

Comando:

```
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"
  }
]
}

```

Per ulteriori informazioni, consulta [Spot Fleet Requests](#) nella Amazon Elastic Compute Cloud User Guide.

- Per API i dettagli, consulta [RequestSpotFleet AWS CLI Command Reference](#).

## request-spot-instances

Il seguente esempio di codice mostra come utilizzare `request-spot-instances`.

### AWS CLI

Per richiedere istanze Spot

Questo comando di esempio crea una richiesta di istanza Spot una tantum per cinque istanze nella zona di disponibilità specificata. Se il tuo account supporta VPC solo la modalità «EC2-», Amazon EC2 avvia le istanze nella sottorete predefinita della zona di disponibilità specificata. Se il tuo account supporta EC2 -Classic, Amazon EC2 avvia le istanze in EC2 -Classic nella zona di disponibilità specificata.

Comando:

```
aws ec2 request-spot-instances --spot-price "0.03" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Specificazione.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"
  }
}
```

### Output:

```
{
  "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"
  },
  ...
]
}

```

Questo comando di esempio crea una richiesta di istanza Spot una tantum per cinque istanze nella sottorete specificata. Amazon EC2 avvia le istanze nella sottorete specificata. Se non VPC è un valore predefinito VPC, per impostazione predefinita le istanze non ricevono un indirizzo IP pubblico.

Comando:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Specificazione.json:

```

{
  "ImageId": "ami-1a2b3c4d",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "SubnetId": "subnet-1a2b3c4d",
  "IamInstanceProfile": {
    "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}

```

Output:

```

{
  "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"
},
...
]
}

```

Questo esempio assegna un indirizzo IP pubblico alle istanze Spot avviate in modo non predefinito. VPC Tieni presente che quando specifichi un'interfaccia di rete, devi includere l'ID di sottorete e l'ID del gruppo di sicurezza utilizzando l'interfaccia di rete.

Comando:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 1 --type "one-time" --launch-specification file://specification.json
```

## Specificazione.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"
  }
}
```

- Per i API dettagli, vedere [RequestSpotInstances](#) in Command Reference.AWS CLI

## reset-address-attribute

Il seguente esempio di codice mostra come utilizzare `reset-address-attribute`.

### AWS CLI

Per reimpostare l'attributo del nome di dominio associato a un indirizzo IP elastico

`reset-address-attribute` Gli esempi seguenti reimpostano l'attributo del nome di dominio di un indirizzo IP elastico.

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
```

Output:

```
{
  "Addresses": [
    {
      "PublicIp": "192.0.2.0",
      "AllocationId": "eipalloc-abcdef01234567890",
      "PtrRecord": "example.com."
      "PtrRecordUpdate": {
        "Value": "example.net.",
        "Status": "PENDING"
      }
    }
  ]
}
```

Per monitorare la modifica in sospeso, consulta [describe-addresses-attribute](#) la sezione AWS CLI Command Reference.

- Per API i dettagli, vedere [ResetAddressAttribute](#) in AWS CLI Command Reference.

## **reset-ebs-default-kms-key-id**

Il seguente esempio di codice mostra come utilizzare `reset-ebs-default-kms-key-id`.

AWS CLI

Per ripristinare le impostazioni predefinite CMK per la EBS crittografia

L'`reset-ebs-default-kms-key-id` esempio seguente reimposta l'impostazione predefinita CMK per EBS la crittografia del tuo AWS account nella regione corrente.

```
aws ec2 reset-ebs-default-kms-key-id
```

Output:

```
{
  "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-a87a-5513eEXAMPLE"
}
```

- Per API i dettagli, vedere [ResetEbsDefaultKmsKeyId](#) in AWS CLI Command Reference.

## reset-fpga-image-attribute

Il seguente esempio di codice mostra come utilizzare `reset-fpga-image-attribute`.

### AWS CLI

Per reimpostare gli attributi di un'FPGA immagine Amazon

Questo esempio reimposta i permessi di caricamento per quanto specificato. AFI

Comando:

```
aws ec2 reset-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --  
attribute LoadPermission
```

Output:

```
{  
  "Return": true  
}
```

- Per API i dettagli, vedere [ResetFpgaImageAttribute](#) in AWS CLI Command Reference.

## reset-image-attribute

Il seguente esempio di codice mostra come utilizzare `reset-image-attribute`.

### AWS CLI

Per reimpostare l' `launchPermission` attributo

Questo esempio reimposta l'`launchPermission` attributo per l'attributo specificato AMI al valore predefinito. Per impostazione predefinita, AMIs sono private. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 reset-image-attribute --image-id ami-5731123e --attribute launchPermission
```

- Per API i dettagli, vedere [ResetImageAttribute](#) in AWS CLI Command Reference.

## reset-instance-attribute

Il seguente esempio di codice mostra come utilizzare `reset-instance-attribute`.

### AWS CLI

Per reimpostare l' `sourceDestCheck` attributo

Questo esempio reimposta l'`sourceDestCheck` attributo dell'istanza specificata. L'istanza deve trovarsi in un VPC. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute sourceDestCheck
```

Per reimpostare l'attributo del kernel

Questo esempio reimposta l'`kernel` attributo dell'istanza specificata. L'istanza deve essere nello stato `stopped`. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute kernel
```

Per reimpostare l'attributo `ramdisk`

Questo esempio reimposta l'`ramdisk` attributo dell'istanza specificata. L'istanza deve essere nello stato `stopped`. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --  
attribute ramdisk
```

- Per API i dettagli, vedere [ResetInstanceAttribute](#) in AWS CLI Command Reference.

## reset-network-interface-attribute

Il seguente esempio di codice mostra come utilizzare `reset-network-interface-attribute`.

## AWS CLI

Per reimpostare un attributo dell'interfaccia di rete

L'`reset-network-interface-attribute` seguente reimposta il valore dell'attributo di controllo source/destination su `true`

```
aws ec2 reset-network-interface-attribute \  
  --network-interface-id eni-686ea200 \  
  --source-dest-check
```

Questo comando non produce alcun output.

- Per i API dettagli, vedere [ResetNetworkInterfaceAttribute](#) in Command Reference.AWS CLI

## `reset-snapshot-attribute`

Il seguente esempio di codice mostra come utilizzare `reset-snapshot-attribute`.

### AWS CLI

Per reimpostare un attributo snapshot

Questo esempio reimposta i permessi di creazione del volume per l'istantanea.

`snap-1234567890abcdef0` Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 reset-snapshot-attribute --snapshot-id snap-1234567890abcdef0 --  
attribute createVolumePermission
```

- Per API i dettagli, vedere [ResetSnapshotAttribute](#) in AWS CLI Command Reference.

## `restore-address-to-classic`

Il seguente esempio di codice mostra come utilizzare `restore-address-to-classic`.

### AWS CLI

Per ripristinare un indirizzo su EC2 -Classic

Questo esempio ripristina l'indirizzo IP elastico 198.51.100.0 sulla piattaforma -Classic. EC2



Comando:

```
aws ec2 restore-address-to-classic --public-ip 198.51.100.0
```

Output:

```
{
  "Status": "MoveInProgress",
  "PublicIp": "198.51.100.0"
}
```

- Per API i dettagli, vedere in Command Reference. [RestoreAddressToClassic](#) AWS CLI

## restore-image-from-recycle-bin

Il seguente esempio di codice mostra come utilizzare `restore-image-from-recycle-bin`.

AWS CLI

Per ripristinare un'immagine dal Cestino

L'`restore-image-from-recycle-bin` esempio seguente ripristina AMI `ami-0111222333444abcd` dal Recycle Bin.

```
aws ec2 restore-image-from-recycle-bin \
  --image-id ami-0111222333444abcd
```

Output:

```
{
  "Return": true
}
```

Per ulteriori informazioni, consulta [Recover AMIs from the Recycle Bin](#) nella Amazon Elastic Compute Cloud User Guide.

- Per API i dettagli, consulta AWS CLI Command [RestoreImageFromRecycleBin](#) Reference.

## restore-managed-prefix-list-version

Il seguente esempio di codice mostra come utilizzare `restore-managed-prefix-list-version`.

## AWS CLI

US-West-2\*\* per ripristinare una versione dell'elenco di prefissi\*\*

Quanto segue `restore-managed-prefix-list-version` ripristina le voci della versione 1 dell'elenco di prefissi specificato.

```
aws ec2 restore-managed-prefix-list-version \
  --prefix-list-id pl-0123456abcabc1 \
  --current-version 2 \
  --previous-version 1
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Managed Prefix Lists](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [RestoreManagedPrefixListVersion AWS CLI Command Reference](#).

## **restore-snapshot-from-recycle-bin**

Il seguente esempio di codice mostra come utilizzare `restore-snapshot-from-recycle-bin`.

### AWS CLI

Per ripristinare le istantanee dal Cestino

L'`restore-snapshot-from-recycle-bin`esempio seguente ripristina un'istanza dal Recycle Bin. Quando si ripristina uno snapshot dal Cestino di riciclaggio, lo snapshot diventa

immediatamente disponibile per l'uso e viene rimosso dal Cestino. Dopo averlo ripristinato, potrà essere utilizzato nello stesso modo in cui qualsiasi altro snapshot viene utilizzato nel proprio account.

```
aws ec2 restore-snapshot-from-recycle-bin \  
  --snapshot-id snap-01234567890abcdef
```

Questo comando non produce alcun output.

Per ulteriori informazioni su Recycle Bin for AmazonEBS, consulta [Recover snapshot from the Recycle Bin](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta Command [RestoreSnapshotFromRecycleBin](#) Reference AWS CLI .

## restore-snapshot-tier

Il seguente esempio di codice mostra come utilizzare `restore-snapshot-tier`.

### AWS CLI

Esempio 1: ripristinare definitivamente un'istantanea archiviata

L'`restore-snapshot-tier` esempio seguente ripristina in modo permanente l'istantanea specificata. Specificate `--snapshot-id` e includete l'`permanent-restore` opzione.

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --permanent-restore
```

Output:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "IsPermanentRestore": true  
}
```

Per ulteriori informazioni sull'archiviazione degli snapshot, consulta [Archive Amazon EBS snapshots](https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/snapshot-archive.html) <[https://docs.aws.amazon.com/AWS\\_EC2/latest/UserGuide/snapshot-archive.html](https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/snapshot-archive.html)> nella Amazon User Guide. EC2

## Esempio 2: per ripristinare temporaneamente uno snapshot archiviato

L'`restore-snapshot-tier` seguente ripristina temporaneamente l'istantanea specificata. Omettere l'opzione `--permanent-restore`. Specificare `--snapshot-id` e, per `temporary-restore-days`, specificare il numero di giorni per i quali ripristinare l'istantanea. `temporary-restore-days` deve essere specificato in giorni. L'intervallo consentito è 1 di 180. Se non si specifica un valore, sarà impostato su 1 giorno.

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --temporary-restore-days 5
```

Output:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "RestoreDuration": 5,  
  "IsPermanentRestore": false  
}
```

Per ulteriori informazioni sull'archiviazione degli snapshot, consulta Archive Amazon EBS snapshots <[https://docs.aws.amazon.com/AWS\\_EC2/latest/UserGuide/snapshot-archive.html](https://docs.aws.amazon.com/AWS_EC2/latest/UserGuide/snapshot-archive.html)> nella Amazon User Guide. EC2

## Esempio 3: modificare il periodo di ripristino

L'`restore-snapshot-tier` seguente modifica il periodo di ripristino per l'istantanea specificata in 10 giorni.

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef  
  --temporary-restore-days 10
```

Output:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "RestoreDuration": 10,  
  "IsPermanentRestore": false  
}
```

Per ulteriori informazioni sull'archiviazione degli snapshot, consulta Archive Amazon EBS snapshots <<https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/snapshot-archive.html>> nella Amazon User Guide. EC2

Esempio 4: per modificare il tipo di ripristino

L'`restore-snapshot-tier` seguente modifica il tipo di ripristino per l'istantanea specificata da temporaneo a permanente.

```
aws ec2 restore-snapshot-tier \  
  --snapshot-id snap-01234567890abcdef \  
  --permanent-restore
```

Output:

```
{  
  "SnapshotId": "snap-01234567890abcdef",  
  "IsPermanentRestore": true  
}
```

Per ulteriori informazioni sull'archiviazione degli snapshot, consulta Archive Amazon EBS snapshots <<https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/snapshot-archive.html>> nella Amazon User Guide. EC2

- Per i dettagli, consulta Command Reference. API [RestoreSnapshotTier](#) AWS CLI

## **revoke-client-vpn-ingress**

Il seguente esempio di codice mostra come utilizzare `revoke-client-vpn-ingress`.

AWS CLI

Per revocare una regola di autorizzazione per un endpoint Client VPN

L'`revoke-client-vpn-ingress` seguente revoca una regola per l'accesso a Internet () `0.0.0.0/0` per tutti i gruppi.

```
aws ec2 revoke-client-vpn-ingress \  
  --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \  
  --target-network-cidr 0.0.0.0/0 --revoke-all-groups
```

**Output:**

```
{
  "Status": {
    "Code": "revoking"
  }
}
```

Per ulteriori informazioni, vedere [Regole di autorizzazione](#) nella AWS Client VPN Administrator Guide.

- Per API i dettagli, vedere [RevokeClientVpnIngress](#) in AWS CLI Command Reference.

**revoke-security-group-egress**

Il seguente esempio di codice mostra come utilizzare `revoke-security-group-egress`.

**AWS CLI**

Esempio 1: rimuovere la regola che consente il traffico in uscita verso un intervallo di indirizzi specifico

Il comando di `revoke-security-group-egress` esempio seguente rimuove la regola che concede l'accesso agli intervalli di indirizzi specificati sulla TCP porta 80.

```
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}]]
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Security groups](#) nella Amazon EC2 User Guide.

Esempio 2: rimuovere la regola che consente il traffico in uscita verso uno specifico gruppo di sicurezza

Il comando di `revoke-security-group-egress` esempio seguente rimuove la regola che concede l'accesso al gruppo di sicurezza specificato sulla TCP porta 80.

```
aws ec2 revoke-security-group-egress \
  --group-id sg-026c12253ce15eff7 \
```

```
--ip-permissions '[{"IpProtocol": "tcp", "FromPort": 443, "ToPort": 443, "UserIdGroupPairs": [{"GroupId": "sg-06df23a01ff2df86d"}]}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Security groups](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [RevokeSecurityGroupEgress AWS CLI Command Reference](#).

## revoke-security-group-ingress

Il seguente esempio di codice mostra come utilizzare `revoke-security-group-ingress`.

### AWS CLI

Esempio 1: rimuovere una regola da un gruppo di sicurezza

L'`revoke-security-group-ingress` seguente rimuove l'accesso alla TCP porta 22 per l'intervallo di `203.0.113.0/24` indirizzi dal gruppo di sicurezza specificato per impostazione predefinita VPC.

```
aws ec2 revoke-security-group-ingress \  
  --group-name mySecurityGroup \  
  --protocol tcp \  
  --port 22 \  
  --cidr 203.0.113.0/24
```

Questo comando non produce alcun output se ha esito positivo.

Per ulteriori informazioni, consulta [Security groups](#) nella Amazon EC2 User Guide.

Esempio 2: rimuovere una regola utilizzando il set di autorizzazioni IP

L'`revoke-security-group-ingress` seguente utilizza il `ip-permissions` parametro per rimuovere una regola in entrata che consente il ICMP messaggio Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (Tipo 3, Codice 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}]
```

Questo comando non produce alcun output se ha esito positivo.

Per ulteriori informazioni, consulta [Security groups](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [RevokeSecurityGroupIngress AWS CLI Command Reference](#).

## run-instances

Il seguente esempio di codice mostra come utilizzare `run-instances`.

### AWS CLI

Esempio 1: per avviare un'istanza in una sottorete predefinita

L'`run-instances` seguente avvia una singola istanza di tipo `t2.micro` nella sottorete predefinita per la regione corrente e la associa alla sottorete predefinita per la regione. VPC La key pair è facoltativa se non prevedi di connetterti alla tua istanza tramite SSH (Linux) o RDP (Windows).

```
aws ec2 run-instances \  
  --image-id ami-0abcdef1234567890 \  
  --instance-type t2.micro \  
  --key-name MyKeyPair
```

Output:

```
{  
  "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-04a636d18e83cfacb",
"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"
}
```

Esempio 2: per avviare un'istanza in una sottorete non predefinita e aggiungere un indirizzo IP pubblico

Nell'esempio di `run-instances` seguente viene richiesto un indirizzo IP pubblico per un'istanza avviata in una sottorete non predefinita. L'istanza è associata al gruppo di sicurezza specificato.

```
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
```

Per un esempio dell'output di `run-instances`, vedi l'Esempio 1.

Esempio 3: per avviare un'istanza con volumi aggiuntivi

Nell'esempio di `run-instances` seguente viene utilizzata una mappatura dei dispositivi a blocchi, specificata in `mapping.json`, per collegare volumi aggiuntivi al momento del lancio. Una mappatura dei dispositivi a blocchi può specificare EBS volumi, volumi di instance store o entrambi i EBS volumi e i volumi di instance store.

```
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
```

Contenuto di `mapping.json`. Questo esempio aggiunge `/dev/sdh` un EBS volume vuoto con una dimensione di 100 GiB.

```
[  
  {  
    "DeviceName": "/dev/sdh",  
    "Ebs": {  
      "VolumeSize": 100  
    }  
  }  
]
```

```
]
```

Contenuto di `mapping.json`. In questo esempio viene aggiunto `ephemeral1`, un volume di archivio dell'istanza.

```
[
  {
    "DeviceName": "/dev/sdc",
    "VirtualName": "ephemeral1"
  }
]
```

Per un esempio dell'output di `run-instances`, vedi l'Esempio 1.

Per ulteriori informazioni sulle mappature dei dispositivi a blocchi, consulta la sezione [Block device mapping](#) nella Amazon EC2 User Guide.

Esempio 4: per avviare un'istanza e aggiungere tag al momento della creazione

Nell'esempio di `run-instances` seguente viene aggiunto un tag con una chiave `webserver` e un valore `production` all'istanza. Il comando applica anche un tag con una chiave `cost-center` e un valore di `cc123` a qualsiasi EBS volume creato (in questo caso, il volume `root`).

```
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}]'
```

Per un esempio dell'output di `run-instances`, vedi l'Esempio 1.

Esempio 5: per avviare un'istanza con dati utente

Nell'esempio di `run-instances` seguente i dati utente vengono trasferiti in un file denominato `my_script.txt` che contiene uno script di configurazione per l'istanza. Lo script viene eseguito al momento dell'avvio.

```
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
```

Per un esempio dell'output di `run-instances`, vedi l'Esempio 1.

Per ulteriori informazioni sui dati utente dell'istanza, consulta [Working with instance user data](#) nella Amazon EC2 User Guide.

Esempio 6: per avviare un'istanza a prestazioni espandibili

Nell'esempio di `run-instances` seguente viene avviata un'istanza `t2.micro` con l'opzione di credito `unlimited`. All'avvio di un'istanza T2, se non si specifica `--credit-specification`, l'opzione di credito predefinita `standard`. All'avvio di un'istanza T3, l'opzione di credito predefinita è `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 CpuCredits=unlimited
```

Per un esempio dell'output di `run-instances`, vedi l'Esempio 1.

Per ulteriori informazioni sulle istanze con prestazioni espandibili, consulta le istanze [Burstable Performance nella](#) Amazon User Guide. EC2

- Per i API dettagli, consulta [RunInstances](#) Command Reference.AWS CLI

## run-scheduled-instances

Il seguente esempio di codice mostra come utilizzare `run-scheduled-instances`.

## AWS CLI

Per avviare un'istanza pianificata

Questo esempio avvia l'istanza pianificata specificata in unVPC.

Comando:

```
aws ec2 run-scheduled-instances --scheduled-instance-  
id sci-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"  
  }  
}
```

Output:

```
{  
  "InstanceIdSet": [  
    "i-1234567890abcdef0"  
  ]  
}
```

Questo esempio avvia l'istanza pianificata specificata in -Classic. EC2

Comando:

```
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"
  }
}
```

Output:

```
{
  "InstanceIdSet": [
    "i-1234567890abcdef0"
  ]
}
```

- Per i API dettagli, vedere in Command Reference. [RunScheduledInstances](#) AWS CLI

## search-local-gateway-routes

Il seguente esempio di codice mostra come utilizzare `search-local-gateway-routes`.

AWS CLI

Per cercare percorsi in una tabella di routing del gateway locale

L'esempio seguente cerca le route statiche nella tabella di routing del gateway locale specificata.

```
aws ec2 search-local-gateway-routes \
```

```
--local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \  
--filters "Name=type,Values=static"
```

Output:

```
{  
  "Route": {  
    "DestinationCidrBlock": "0.0.0.0/0",  
    "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",  
    "Type": "static",  
    "State": "deleted",  
    "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"  
  }  
}
```

- Per API i dettagli, vedere [SearchLocalGatewayRoutes](#) in AWS CLI Command Reference.

## search-transit-gateway-multicast-groups

Il seguente esempio di codice mostra come utilizzare `search-transit-gateway-multicast-groups`.

AWS CLI

Per cercare uno o più gruppi multicast di Transit Gateway e restituire le informazioni sull'appartenenza al gruppo

L'`search-transit-gateway-multicast-groups` esempio seguente restituisce l'appartenenza al gruppo multicast specificato.

```
aws ec2 search-transit-gateway-multicast-groups \  
--transit-gateway-multicast-domain-id tgw-mcast-domain-000fb24d04EXAMPLE
```

Output:

```
{  
  "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"
    }
]
}

```

Per ulteriori informazioni, vedere [Managing multicast group](#) nella Transit Gateways Guide.

- Per API i dettagli, vedere [SearchTransitGatewayMulticastGroups](#) in AWS CLI Command Reference.

## search-transit-gateway-routes

Il seguente esempio di codice mostra come utilizzare `search-transit-gateway-routes`.

### AWS CLI

Per cercare percorsi nella tabella delle rotte del gateway di transito specificata

L'`search-transit-gateway-routes` esempio seguente restituisce tutti i percorsi che sono di tipo `static` nella tabella delle rotte specificata.

```

aws ec2 search-transit-gateway-routes \
  --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
  --filters "Name=type,Values=static"

```

Output:

```

{
  "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
}
```

Per ulteriori informazioni, consulta le [tabelle delle rotte dei gateway](#) di transito nella Guida ai gateway di transito.

- Per API i dettagli, vedere [SearchTransitGatewayRoutes](#) in AWS CLI Command Reference.

## send-diagnostic-interrupt

Il seguente esempio di codice mostra come utilizzare `send-diagnostic-interrupt`.

### AWS CLI

Per inviare un'interruzione diagnostica

L'esempio seguente invia un'interruzione diagnostica all'istanza specificata.

```
aws ec2 send-diagnostic-interrupt \
  --instance-id i-1234567890abcdef0
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [SendDiagnosticInterrupt](#) in AWS CLI Command Reference.

## start-instances

Il seguente esempio di codice mostra come utilizzare `start-instances`.

### AWS CLI

Per avviare un'EC2istanza Amazon

Questo esempio avvia l'istanza EBS supportata da Amazon specificata.

Comando:

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

Output:

```
{
  "StartingInstances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "CurrentState": {
        "Code": 0,
        "Name": "pending"
      },
      "PreviousState": {
        "Code": 80,
        "Name": "stopped"
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta [Arrestare e avviare un'istanza](#) nella Guida per l'utente di Amazon Elastic Compute Cloud.

- Per API i dettagli, consulta [StartInstances AWS CLI Command Reference](#).

## start-network-insights-access-scope-analysis

Il seguente esempio di codice mostra come utilizzare `start-network-insights-access-scope-analysis`.

## AWS CLI

Per avviare un'analisi dell'ambito di accesso a Network Insights

L'`start-network-insights-access-scope-analysis` esempio seguente avvia l'analisi dell'ambito nel tuo AWS account.

```
aws ec2 start-network-insights-access-scope-analysis \
  --region us-east-1 \
  --network-insights-access-scope-id nis-123456789111
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Guida introduttiva all'utilizzo di Network Access Analyzer utilizzando la AWS CLI](#) guida Network Access Analyzer.

- Per API i dettagli, vedere [StartNetworkInsightsAccessScopeAnalysis](#) in AWS CLI Command Reference.

## **start-network-insights-analysis**

Il seguente esempio di codice mostra come utilizzare `start-network-insights-analysis`.

## AWS CLI

Per analizzare un percorso

L'`start-network-insights-analysis` esempio seguente analizza il percorso tra l'origine e la destinazione. Per visualizzare i risultati dell'analisi del percorso, utilizzate il `describe-network-insights-analysis` comando.

```
aws ec2 start-network-insights-analysis \  
  --network-insights-path-id nip-0b26f224f1d131fa8
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta la [Guida introduttiva al AWS CLI Reachability Analyzer Guide](#).

- Per API i dettagli, vedere [StartNetworkInsightsAnalysis](#) in AWS CLI Command Reference.

## start-vpc-endpoint-service-private-dns-verification

Il seguente esempio di codice mostra come utilizzare `start-vpc-endpoint-service-private-dns-verification`.

### AWS CLI

Per avviare il processo di DNS verifica

L'`start-vpc-endpoint-service-private-dns-verification` esempio seguente avvia il processo di DNS verifica per il servizio endpoint specificato.

```
aws ec2 start-vpc-endpoint-service-private-dns-verification \  
  --service-id vpce-svc-071afff70666e61e0
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Gestire i DNS nomi nella Guida](#) per l'AWS PrivateLink utente.

- Per API i dettagli, vedere [StartVpcEndpointServicePrivateDnsVerification](#) in AWS CLI Command Reference.

## stop-instances

Il seguente esempio di codice mostra come utilizzare `stop-instances`.

### AWS CLI

Esempio 1: interrompere un'EC2istanza Amazon

L'`stop-instances`esempio seguente interrompe l'istanza EBS supportata da Amazon specificata.

```
aws ec2 stop-instances \  
  --instance-ids i-1234567890abcdef0
```

Output:

```
{  
  "StoppingInstances": [  
    {  
      "InstanceId": "i-1234567890abcdef0",  
      "CurrentState": {  
        "Code": 64,  
        "Name": "stopping"  
      },  
      "PreviousState": {  
        "Code": 16,  
        "Name": "running"  
      }  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Arrestare e avviare un'istanza](#) nella Guida per l'utente di Amazon Elastic Compute Cloud.

Esempio 2: ibernare un'istanza Amazon EC2

L'`stop-instances`esempio seguente mette in ibernazione l'istanza supportata da EBS Amazon se l'istanza è abilitata per l'ibernazione e soddisfa i prerequisiti di ibernazione. Dopo l'ibernazione dell'istanza, questa viene arrestata.

```
aws ec2 stop-instances \  
  --instance-ids i-1234567890abcdef0
```

```
--instance-ids i-1234567890abcdef0 \  
--hibernate
```

Output:

```
{  
  "StoppingInstances": [  
    {  
      "CurrentState": {  
        "Code": 64,  
        "Name": "stopping"  
      },  
      "InstanceId": "i-1234567890abcdef0",  
      "PreviousState": {  
        "Code": 16,  
        "Name": "running"  
      }  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [ibernazione di un'istanza Linux on demand](#) nella Guida per l'utente di Amazon Elastic Cloud Compute.

- API Per i dettagli [StopInstances AWS CLI](#), consulta Command Reference.

## terminate-client-vpn-connections

Il seguente esempio di codice mostra come utilizzare `terminate-client-vpn-connections`.

AWS CLI

Per terminare una connessione a un endpoint Client VPN

L'`terminate-client-vpn-connection`esempio seguente termina la connessione specificata all'endpoint Client. VPN

```
aws ec2 terminate-client-vpn-connections \  
--client-vpn-endpoint-id vpn-endpoint-123456789123abcde \  
--connection-id cvpn-connection-04edd76f5201e0cb8
```

Output:

```
{
  "ClientVpnEndpointId": "vpn-endpoint-123456789123abcde",
  "ConnectionStatuses": [
    {
      "ConnectionId": "cvpn-connection-04edd76f5201e0cb8",
      "PreviousStatus": {
        "Code": "active"
      },
      "CurrentStatus": {
        "Code": "terminating"
      }
    }
  ]
}
```

Per ulteriori informazioni, vedere [Client Connections nella Client VPN Administrator AWS Guide](#).

- Per API i dettagli, vedere [TerminateClientVpnConnections](#) in AWS CLI Command Reference.

## terminate-instances

Il seguente esempio di codice mostra come utilizzare `terminate-instances`.

### AWS CLI

Per terminare un'istanza Amazon EC2

Questo esempio termina l'istanza specificata.

Comando:

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

Output:

```
{
  "TerminatingInstances": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "CurrentState": {
        "Code": 32,
```



```

        "Name": "shutting-down"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Using Amazon EC2 Instances](#) nella [AWS Command Line Interface User Guide](#).

- Per API i dettagli, consulta [TerminateInstances AWS CLI Command Reference](#).

## unassign-ipv6-addresses

Il seguente esempio di codice mostra come utilizzare `unassign-ipv6-addresses`.

### AWS CLI

Per annullare l'assegnazione di un IPv6 indirizzo da un'interfaccia di rete

Questo esempio annulla l'assegnazione dell'IPv6 indirizzo specificato dall'interfaccia di rete specificata.

Comando:

```
aws ec2 unassign-ipv6-addresses --ipv6-
addresses 2001:db8:1234:1a00:3304:8879:34cf:4071 --network-interface-id eni-23c49b68
```

Output:

```
{
  "NetworkInterfaceId": "eni-23c49b68",
  "UnassignedIpv6Addresses": [
    "2001:db8:1234:1a00:3304:8879:34cf:4071"
  ]
}
```

- Per API i dettagli, vedere [UnassignIpv6Addresses](#) in [AWS CLI Command Reference](#).

## unassign-private-ip-addresses

Il seguente esempio di codice mostra come utilizzare `unassign-private-ip-addresses`.

### AWS CLI

Per annullare l'assegnazione di un indirizzo IP privato secondario da un'interfaccia di rete

Questo esempio annulla l'assegnazione dell'indirizzo IP privato specificato dall'interfaccia di rete specificata. Se il comando va a buon fine, non viene restituito alcun output.

Comando:

```
aws ec2 unassign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-ip-addresses 10.0.0.82
```

- Per API i dettagli, vedere [UnassignPrivateIpAddresses](#) in AWS CLI Command Reference.

## unassign-private-nat-gateway-address

Il seguente esempio di codice mostra come utilizzare `unassign-private-nat-gateway-address`.

### AWS CLI

Per annullare l'assegnazione di un indirizzo IP privato dal gateway privato NAT

L'esempio seguente annulla l'assegnazione dell'indirizzo IP specificato al gateway privato specificato. NAT

```
aws ec2 unassign-private-nat-gateway-address \  
--nat-gateway-id nat-1234567890abcdef0 \  
--private-ip-addresses 10.0.20.197
```

Output:

```
{  
  "NatGatewayId": "nat-0ee3edd182361f662",  
  "NatGatewayAddresses": [  
    {  
      "NetworkInterfaceId": "eni-0065a61b324d1897a",
```

```
        "PrivateIp": "10.0.20.197",
        "IsPrimary": false,
        "Status": "unassigning"
    }
]
}
```

Per ulteriori informazioni, consulta i [NATgateway](#) nella Amazon VPC User Guide.

- Per API i dettagli, consulta [UnassignPrivateNatGatewayAddress AWS CLI Command Reference](#).

## unlock-snapshot

Il seguente esempio di codice mostra come utilizzare `unlock-snapshot`.

### AWS CLI

Per sbloccare un'istantanea

L'`unlock-snapshot` esempio seguente sblocca l'istantanea specificata.

```
aws ec2 unlock-snapshot \
  --snapshot-id snap-0b5e733b4a8df6e0d
```

Output:

```
{
  "SnapshotId": "snap-0b5e733b4a8df6e0d"
}
```

Per ulteriori informazioni, consulta [Snapshot lock](#) nella Amazon EBS User Guide.

- Per API i dettagli, consulta [UnlockSnapshot AWS CLI Command Reference](#).

## unmonitor-instances

Il seguente esempio di codice mostra come utilizzare `unmonitor-instances`.

### AWS CLI

Per disabilitare il monitoraggio dettagliato per un'istanza

Questo comando di esempio disabilita il monitoraggio dettagliato per l'istanza specificata.

Comando:

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

Output:

```
{
  "InstanceMonitorings": [
    {
      "InstanceId": "i-1234567890abcdef0",
      "Monitoring": {
        "State": "disabling"
      }
    }
  ]
}
```

- Per API i dettagli, vedere [UnmonitorInstances](#) in AWS CLI Command Reference.

## update-security-group-rule-descriptions-egress

Il seguente esempio di codice mostra come utilizzare `update-security-group-rule-descriptions-egress`.

AWS CLI

Per aggiornare la descrizione di una regola del gruppo di sicurezza in uscita

L'esempio seguente aggiorna la descrizione della regola del gruppo di sicurezza per la porta e l'intervallo di IPv4 indirizzi specificati. La descrizione 'Outbound HTTP access to server 2' sostituisce qualsiasi descrizione esistente per la regola.

```
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"}]
```

Output:

```
{
  "Return": true
}
```

Per ulteriori informazioni, consulta [le regole dei gruppi di sicurezza](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [UpdateSecurityGroupRuleDescriptionsEgress AWS CLI Command Reference](#).

## update-security-group-rule-descriptions-ingress

Il seguente esempio di codice mostra come utilizzare `update-security-group-rule-descriptions-ingress`.

AWS CLI

Esempio 1: aggiornare la descrizione di una regola del gruppo di sicurezza in entrata con una fonte CIDR

L'`update-security-group-rule-descriptions-ingress` seguente aggiorna la descrizione della regola del gruppo di sicurezza per la porta e l'intervallo di IPv4 indirizzi specificati. La descrizione 'SSH access from ABC office' sostituisce qualsiasi descrizione esistente per la regola.

```
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"}]'
```

Output:

```
{
  "Return": true
}
```

Per ulteriori informazioni, consulta [le regole dei gruppi di sicurezza](#) nella Amazon EC2 User Guide.

Esempio 2: aggiornare la descrizione di una regola del gruppo di sicurezza in entrata con una fonte con un elenco di prefissi

L'`update-security-group-rule-descriptions-ingress` seguente aggiorna la descrizione della regola del gruppo di sicurezza per l'elenco di porte e prefissi specificato. La descrizione 'SSH access from ABC office' sostituisce qualsiasi descrizione esistente per la regola.

```
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"}]'
```

Output:

```
{
  "Return": true
}
```

Per ulteriori informazioni, consulta [le regole dei gruppi di sicurezza](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [UpdateSecurityGroupRuleDescriptionsIngress AWS CLI Command Reference](#).

## **withdraw-byoip-cidr**

Il seguente esempio di codice mostra come utilizzare `withdraw-byoip-cidr`.

AWS CLI

Per interrompere la pubblicità di un intervallo di indirizzi

L'`withdraw-byoip-cidr` seguente interrompe la pubblicità dell'intervallo di indirizzi specificato.

```
aws ec2 withdraw-byoip-cidr
  --cidr 203.0.113.25/24
```

Output:

```
{
  "ByoipCidr": {
    "Cidr": "203.0.113.25/24",
    "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
    "State": "advertised"
  }
}
```

- Per API i dettagli, vedere [WithdrawByoipCidr](#) in AWS CLI Command Reference.

## Esempi di Amazon EC2 Instance Connect con AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon EC2 Instance Connect. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **send-ssh-public-key**

Il seguente esempio di codice mostra come utilizzare `send-ssh-public-key`.

#### AWS CLI

Per inviare una chiave SSH pubblica a un'istanza

L'esempio seguente invia la chiave SSH pubblica specificata all'istanza specificata. La chiave viene utilizzata per autenticare l'utente specificato.

```
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
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [SendSshPublicKey](#) in AWS CLI Command Reference.

## ECREsempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con Amazon ECR.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **batch-check-layer-availability**

Il seguente esempio di codice mostra come utilizzare `batch-check-layer-availability`.

AWS CLI

Per verificare la disponibilità di un livello

L'esempio seguente verifica la disponibilità di un layer con il digest

`sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed`  
nel `cluster-autoscaler` repository.

```
aws ecr batch-check-layer-availability \  
--repository-name cluster-autoscaler \  

```



```
--layer-
digests sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
```

Output:

```
{
  "layers": [
    {
      "layerDigest":
"sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed",
      "layerAvailability": "AVAILABLE",
      "layerSize": 2777,
      "mediaType": "application/vnd.docker.container.image.v1+json"
    }
  ],
  "failures": []
}
```

- Per API i dettagli, vedere [BatchCheckLayerAvailability](#) in AWS CLI Command Reference.

## batch-delete-image

Il seguente esempio di codice mostra come utilizzare `batch-delete-image`.

### AWS CLI

Esempio 1: eliminare un'immagine

L'`batch-delete-image` seguente elimina un'immagine con il tag `precise` nel repository specificato nel registro predefinito di un account.

```
aws ecr batch-delete-image \
  --repository-name ubuntu \
  --image-ids imageTag=precise
```

Output:

```
{
  "failures": [],
  "imageIds": [
    {
      "imageTag": "precise",
```

```

        "imageDigest":
          "sha256:19665f1e6d1e504117a1743c0a3d3753086354a38375961f2e665416ef4b1b2f"
      }
    ]
  }

```

Esempio 2: Per eliminare più immagini

L'`batch-delete-image` seguente elimina tutte le immagini contrassegnate con `prod` e `team1` presenti nel repository specificato.

```

aws ecr batch-delete-image \
  --repository-name MyRepository \
  --image-ids imageTag=prod imageTag=team1

```

Output:

```

{
  "imageIds": [
    {
      "imageDigest": "sha256:123456789012",
      "imageTag": "prod"
    },
    {
      "imageDigest": "sha256:567890121234",
      "imageTag": "team1"
    }
  ],
  "failures": []
}

```

Per ulteriori informazioni, consulta [Eliminazione di un'immagine](#) nella Amazon ECR User Guide.

- Per API i dettagli, consulta [BatchDeleteImage AWS CLI Command Reference](#).

## batch-get-image

Il seguente esempio di codice mostra come utilizzare `batch-get-image`.

### AWS CLI

Esempio 1: Per ottenere un'immagine

L'batch-get-image seguente ottiene un'immagine con il tag v1.13.6 in un repository chiamato cluster-autoscaler nel registro predefinito di un account.

```
aws ecr batch-get-image \
  --repository-name cluster-autoscaler \
  --image-ids imageTag=v1.13.6
```

Output:

```
{
  "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\":
\"sha256:39fafc05754f195f134ca11ecdb1c9a691ab0848c697fffeb5a85f900caaf6e1\"\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": []
  ]
}
```

```
}
```

Esempio 2: per ottenere più immagini

L'batch-get-imageesempio seguente mostra i dettagli di tutte le immagini etichettate con prod e team1 presenti nel repository specificato.

```
aws ecr batch-get-image \  
  --repository-name MyRepository \  
  --image-ids imageTag=prod imageTag=team1
```

Output:

```
{  
  "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": []  
}
```

Per ulteriori informazioni, consulta [le immagini](#) nella Amazon ECR User Guide.

- Per API i dettagli, consulta [BatchGetImage AWS CLI Command Reference](#).

## complete-layer-upload

Il seguente esempio di codice mostra come utilizzare `complete-layer-upload`.

### AWS CLI

Per completare il caricamento di un livello di immagine

L'esempio seguente completa il caricamento di un livello di immagine nel `layer-test` repository.

```
aws ecr complete-layer-upload \  
  --repository-name layer-test \  
  --upload-id 6cb64b8a-9378-0e33-2ab1-b780fab8a9e9 \  
  --layer-digests 6cb64b8a-9378-0e33-2ab1-  
b780fab8a9e9:48074e6d3a68b39aad8ccc002cdad912d4148c0f92b3729323e
```

Output:

```
{  
  "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",  
  "layerDigest":  
    "sha256:9a77f85878aa1906f2020a0ecdf7a7e962d57e882250acd773383224b3fe9a02",  
  "repositoryName": "layer-test",  
  "registryId": "130757420319"  
}
```

- Per API i dettagli, vedere [CompleteLayerUpload](#) in AWS CLI Command Reference.

## create-repository

Il seguente esempio di codice mostra come utilizzare `create-repository`.

### AWS CLI

Esempio 1: creare un repository

L'esempio seguente crea un repository all'interno dello spazio dei nomi specificato nel registro predefinito per un account.

```
aws ecr create-repository \  
  --repository-name layer-test
```

```
--repository-name project-a/sample-repo
```

Output:

```
{
  "repository": {
    "registryId": "123456789012",
    "repositoryName": "project-a/sample-repo",
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/sample-repo"
  }
}
```

Per ulteriori informazioni, consulta [Creating a Repository](#) nella Amazon ECR User Guide.

Esempio 2: creare un repository configurato con l'immutabilità dei tag di immagine

L'`create-repository` seguente crea un repository configurato per l'immutabilità dei tag nel registro predefinito di un account.

```
aws ecr create-repository \
  --repository-name project-a/sample-repo \
  --image-tag-mutability IMMUTABLE
```

Output:

```
{
  "repository": {
    "registryId": "123456789012",
    "repositoryName": "project-a/sample-repo",
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/sample-repo",
    "imageTagMutability": "IMMUTABLE"
  }
}
```

Per ulteriori informazioni, consulta [Image Tag Mutability](#) nella Amazon ECR User Guide.

Esempio 3: creare un repository configurato con una configurazione di scansione

L'`create-repository` seguente crea un repository configurato per eseguire una scansione di vulnerabilità su image push nel registro predefinito di un account.

```
aws ecr create-repository \  
  --repository-name project-a/sample-repo \  
  --image-scanning-configuration scanOnPush=true
```

Output:

```
{  
  "repository": {  
    "registryId": "123456789012",  
    "repositoryName": "project-a/sample-repo",  
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/  
sample-repo",  
    "imageScanningConfiguration": {  
      "scanOnPush": true  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Image Scanning](#) nella Amazon ECR User Guide.

- Per API i dettagli, consulta [CreateRepository AWS CLI Command Reference](#).

## delete-lifecycle-policy

Il seguente esempio di codice mostra come utilizzare `delete-lifecycle-policy`.

AWS CLI

Per eliminare la politica del ciclo di vita di un repository

L'esempio seguente elimina la politica del ciclo di vita per il repository `hello-world`

```
aws ecr delete-lifecycle-policy \  
  --repository-name hello-world
```

Output:

```
{  
  "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
}

```

- Per i API dettagli, vedere [DeleteLifecyclePolicy](#) in Command Reference.AWS CLI

## delete-repository-policy

Il seguente esempio di codice mostra come utilizzare `delete-repository-policy`.

### AWS CLI

Per eliminare la politica di repository per un repository

L'esempio seguente elimina la policy di repository per il repository `cluster-autoscaler`

```

aws ecr delete-repository-policy \
  --repository-name cluster-autoscaler

```

Output:

```

{
  "registryId": "012345678910",
  "repositoryName": "cluster-autoscaler",
  "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
  \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :
  \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage
  \", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
}

```

- Per API i dettagli, vedere [DeleteRepositoryPolicy](#) in AWS CLI Command Reference.

## delete-repository

Il seguente esempio di codice mostra come utilizzare `delete-repository`.



## AWS CLI

Per eliminare un repository

L'`delete-repository` seguente comando forza l'eliminazione del repository specificato nel registro predefinito per un account. Il `--force` flag è obbligatorio se il repository contiene immagini.

```
aws ecr delete-repository \  
  --repository-name ubuntu \  
  --force
```

Output:

```
{  
  "repository": {  
    "registryId": "123456789012",  
    "repositoryName": "ubuntu",  
    "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/ubuntu"  
  }  
}
```

Per ulteriori informazioni, consulta [Eliminazione di un repository](#) nella Amazon ECR User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteRepository](#) Reference.

## describe-image-scan-findings

Il seguente esempio di codice mostra come utilizzare `describe-image-scan-findings`.

### AWS CLI

Per descrivere i risultati della scansione di un'immagine

L'`describe-image-scan-findings` seguente comando restituisce i risultati della scansione di un'immagine utilizzando l'immagine digest nel repository specificato nel registro predefinito di un account.

```
aws ecr describe-image-scan-findings \  
  --repository-name sample-repo \  
  --image-  
id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6
```

## Output:

```
{
  "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."
  }
}
```

Per ulteriori informazioni, consulta [Image Scanning](#) nella Amazon ECR User Guide.

- Per API i dettagli, consulta [DescribeImageScanFindings AWS CLI Command Reference](#).

## describe-images

Il seguente esempio di codice mostra come utilizzare `describe-images`.

### AWS CLI

Descrivere un'immagine in un repository

L'esempio seguente mostra i dettagli di un'immagine nel `cluster-autoscaler` repository con il tag `v1.13.6`

```
aws ecr describe-images \
  --repository-name cluster-autoscaler \
  --image-ids imageTag=v1.13.6
```

Output:

```
{
  "imageDetails": [
    {
      "registryId": "012345678910",
      "repositoryName": "cluster-autoscaler",
      "imageDigest":
"sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
      "imageTags": [
        "v1.13.6"
      ],
      "imageSizeInBytes": 48318255,
      "imagePushedAt": 1565128275.0
    }
  ]
}
```

```
}
```

- Per API i dettagli, vedere [DescribeImages](#) in AWS CLI Command Reference.

## describe-repositories

Il seguente esempio di codice mostra come utilizzare `describe-repositories`.

### AWS CLI

Per descrivere i repository in un registro

Questo esempio descrive i repository nel registro predefinito per un account.

Comando:

```
aws ecr describe-repositories
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeRepositories](#) in AWS CLI Command Reference.

## get-authorization-token

Il seguente esempio di codice mostra come utilizzare `get-authorization-token`.

## AWS CLI

Per ottenere un token di autorizzazione per il registro predefinito

Il comando di `get-authorization-token` esempio seguente ottiene un token di autorizzazione per il registro predefinito.

```
aws ecr get-authorization-token
```

Output:

```
{
  "authorizationData": [
    {
      "authorizationToken": "QVdT0kN...",
      "expiresAt": 1448875853.241,
      "proxyEndpoint": "https://123456789012.dkr.ecr.us-west-2.amazonaws.com"
    }
  ]
}
```

- Per API i dettagli, vedere [GetAuthorizationToken](#) in AWS CLI Command Reference.

## get-download-url-for-layer

Il seguente esempio di codice mostra come utilizzare `get-download-url-for-layer`.

### AWS CLI

Per scaricare URL un layer

L'`get-download-url-for-layer` esempio seguente mostra il download URL di un layer con il digest

`sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed`  
nel `cluster-autoscaler` repository.

```
aws ecr get-download-url-for-layer \
  --repository-name cluster-autoscaler \
  --layer-  
digest sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
```

**Output:**

```
{
  "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"
}
```

- Per API i dettagli, vedete [GetDownloadUrlForLayer](#) in AWS CLI Command Reference.

**get-lifecycle-policy-preview**

Il seguente esempio di codice mostra come utilizzare `get-lifecycle-policy-preview`.

**AWS CLI**

Per recuperare i dettagli per un'anteprima della politica del ciclo di vita

L'`get-lifecycle-policy-preview` seguente recupera il risultato di un'anteprima dei criteri del ciclo di vita per l'archivio specificato nel registro predefinito di un account.

Comando:

```
aws ecr get-lifecycle-policy-preview \
  --repository-name "project-a/amazon-ecs-sample"
```

**Output:**

```
{
  "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  ],\n}
```

```

  \"action\": {\n
    \n
    \"type\": \"expire\"\n
  }\n
  }\n
  ]\n}\n",
  "status": "COMPLETE",
  "previewResults": [],
  "summary": {
    "expiringImageTotalCount": 0
  }
}

```

Per ulteriori informazioni, consulta [Lifecycle Policies](#) nella Amazon ECR User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetLifecyclePolicyPreview](#)Reference.

## get-lifecycle-policy

Il seguente esempio di codice mostra come utilizzare `get-lifecycle-policy`.

### AWS CLI

Per recuperare una politica del ciclo di vita

L'esempio seguente visualizza i dettagli della politica del ciclo di vita per il repository specificato nel registro predefinito per l'account.

```

aws ecr get-lifecycle-policy \
  --repository-name "project-a/amazon-ecs-sample"

```

Output:

```

{
  "registryId": "123456789012",
  "repositoryName": "project-a/amazon-ecs-sample",
  "lifecyclePolicyText": "{\n  \"rules\": [\n    {\n      \"rulePriority\": 1,\n      \"description\":\n        \"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}",
  "lastEvaluatedAt": 1504295007.0
}

```

Per ulteriori informazioni, consulta [Lifecycle Policies](#) nella Amazon ECR User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetLifecyclePolicy](#)Reference.

## get-login-password

Il seguente esempio di codice mostra come utilizzare `get-login-password`.

### AWS CLI

Per recuperare una password per l'autenticazione in un registro

Quanto segue `get-login-password` mostra una password che puoi utilizzare con un client container di tua scelta per autenticarti in qualsiasi ECR registro Amazon a cui il tuo IAM principale ha accesso.

```
aws ecr get-login-password
```

Output:

```
<password>
```

Da utilizzare con DockerCLI, reindirizza l'output del `get-login-password` comando al `docker login` comando. Quando recuperi la password, assicurati di specificare la stessa regione in cui si trova il tuo ECR registro Amazon.

```
aws ecr get-login-password \  
  --region <region> \  
 \  
| docker login \  
  --username AWS \  
  --password-stdin <aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

Per ulteriori informazioni, consulta [Registry Authentication](#) nella Amazon ECR User Guide.

- Per API i dettagli, consulta [GetLoginPassword AWS CLI](#) Command Reference.

## get-login

Il seguente esempio di codice mostra come utilizzare `get-login`.

### AWS CLI

Per recuperare un comando di accesso Docker nel registro predefinito



Questo esempio stampa un comando che puoi usare per accedere al tuo ECR registro Amazon predefinito.

Comando:

```
aws ecr get-login
```

Output:

```
docker login -u AWS -p <password> -e none https://  
<aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

Per accedere al registro di un altro account

Questo esempio stampa uno o più comandi che puoi usare per accedere ai ECR registri Amazon associati ad altri account.

Comando:

```
aws ecr get-login --registry-ids 012345678910 023456789012
```

Output:

```
docker login -u <username> -p <token-1> -e none <endpoint-1>  
docker login -u <username> -p <token-2> -e none <endpoint-2>
```

- Per API i dettagli, consulta AWS CLI Command [GetLoginReference](#).

## get-repository-policy

Il seguente esempio di codice mostra come utilizzare `get-repository-policy`.

AWS CLI

Per recuperare la politica di repository per un repository

L'esempio seguente visualizza i dettagli sulla politica di repository per il repository `cluster-autoscaler`

```
aws ecr get-repository-policy \
```

```
--repository-name cluster-autoscaler
```

Output:

```
{
  "registryId": "012345678910",
  "repositoryName": "cluster-autoscaler",
  "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n    \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" : \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
```

- Per API i dettagli, vedere [GetRepositoryPolicy](#) in AWS CLI Command Reference.

## **initiate-layer-upload**

Il seguente esempio di codice mostra come utilizzare `initiate-layer-upload`.

AWS CLI

Per avviare il caricamento di un livello di immagine

L'esempio seguente avvia il caricamento di un livello di immagine nel `layer-test` repository.

```
aws ecr initiate-layer-upload \  
  --repository-name layer-test
```

Output:

```
{
  "partSize": 10485760,
  "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9"
}
```

- Per API i dettagli, vedere [InitiateLayerUpload](#) in AWS CLI Command Reference.

## **list-images**

Il seguente esempio di codice mostra come utilizzare `list-images`.

## AWS CLI

Per elencare le immagini in un repository

L'`list-images` seguente visualizza un elenco delle immagini presenti nel `cluster-autoscaler` repository.

```
aws ecr list-images \
  --repository-name cluster-autoscaler
```

Output:

```
{
  "imageIds": [
    {
      "imageDigest":
"sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",
      "imageTag": "v1.13.8"
    },
    {
      "imageDigest":
"sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",
      "imageTag": "v1.13.7"
    },
    {
      "imageDigest":
"sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
      "imageTag": "v1.13.6"
    }
  ]
}
```

- Per API i dettagli, vedere [ListImages](#) in AWS CLI Command Reference.

## **list-tags-for-resource**

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

## AWS CLI

Per elencare i tag per il repository

L'`list-tags-for-resource` seguente visualizza un elenco dei tag associati al `hello-world` repository.

```
aws ecr list-tags-for-resource \  
  --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world
```

Output:

```
{  
  "tags": [  
    {  
      "Key": "Stage",  
      "Value": "Integ"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## `put-image-scanning-configuration`

Il seguente esempio di codice mostra come utilizzare `put-image-scanning-configuration`.

AWS CLI

Per aggiornare la configurazione di scansione delle immagini per un repository

L'`put-image-scanning-configuration` seguente aggiorna la configurazione di scansione delle immagini per il repository specificato.

```
aws ecr put-image-scanning-configuration \  
  --repository-name sample-repo \  
  --image-scanning-configuration scanOnPush=true
```

Output:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "sample-repo",  
  "imageScanningConfiguration": {  
    "scanOnPush": true  
  }  
}
```

```
}  
}
```

Per ulteriori informazioni, consulta [Image Scanning](#) nella Amazon ECR User Guide.

- Per API i dettagli, consulta [PutImageScanningConfiguration AWS CLI Command Reference](#).

## **put-image-tag-mutability**

Il seguente esempio di codice mostra come utilizzare `put-image-tag-mutability`.

### AWS CLI

Per aggiornare l'impostazione di mutabilità del tag di immagine per un repository

L'`put-image-tag-mutability` esempio seguente configura il repository specificato per l'immutabilità dei tag. Ciò impedisce la sovrascrittura di tutti i tag di immagine all'interno del repository.

```
aws ecr put-image-tag-mutability \  
  --repository-name hello-repository \  
  --image-tag-mutability IMMUTABLE
```

Output:

```
{  
  "registryId": "012345678910",  
  "repositoryName": "sample-repo",  
  "imageTagMutability": "IMMUTABLE"  
}
```

Per ulteriori informazioni, consulta [Image Tag Mutability](#) nella Amazon ECR User Guide.

- Per API i dettagli, consulta [PutImageTagMutability AWS CLI Command Reference](#).

## **put-image**

Il seguente esempio di codice mostra come utilizzare `put-image`.

### AWS CLI

Per rietichettare un'immagine con il relativo manifesto

L'put-image esempio seguente crea un nuovo tag nel hello-world repository con un manifesto di immagini esistente.

```
aws ecr put-image \  
  --repository-name hello-world \  
  --image-tag 2019.08 \  
  --image-manifest file://hello-world.manifest.json
```

Contenuto di 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"
    }
  ]
```

}

## Output:

```
{
  "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\ ":
\n      \ "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      }
}

```

- Per API i dettagli, vedere [PutImage](#) in AWS CLI Command Reference.

## put-lifecycle-policy

Il seguente esempio di codice mostra come utilizzare `put-lifecycle-policy`.

### AWS CLI

Per creare una politica del ciclo di vita

L'`put-lifecycle-policy` esempio seguente crea una politica del ciclo di vita per il repository specificato nel registro predefinito per un account.

```

aws ecr put-lifecycle-policy \
  --repository-name "project-a/amazon-ecs-sample" \
  --lifecycle-policy-text "file://policy.json"

```

Contenuto di `policy.json`.

```

{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Expire images older than 14 days",
      "selection": {
        "tagStatus": "untagged",
        "countType": "sinceImagePushed",
        "countUnit": "days",
        "countNumber": 14
      }
    }
  ],
}

```

```

        "action": {
            "type": "expire"
        }
    ]
}

```

Output:

```

{
  "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\"}}]}"
}

```

Per ulteriori informazioni, consulta [Lifecycle Policies](#) nella Amazon ECR User Guide.

- Per API i dettagli, consulta AWS CLI Command [PutLifecyclePolicy](#) Reference.

## set-repository-policy

Il seguente esempio di codice mostra come utilizzare `set-repository-policy`.

### AWS CLI

Per impostare la politica di repository per un repository

L'esempio seguente allega al repository una policy di repository contenuta in un file. `cluster-autoscaler`

```

aws ecr set-repository-policy \
  --repository-name cluster-autoscaler \
  --policy-text file://my-policy.json

```

Contenuto di `my-policy.json`.

```

{
  "Version" : "2008-10-17",
  "Statement" : [

```

```

    {
      "Sid" : "allow public pull",
      "Effect" : "Allow",
      "Principal" : "*",
      "Action" : [
        "ecr:BatchCheckLayerAvailability",
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer"
      ]
    }
  ]
}

```

Output:

```

{
  "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}"
}

```

- Per API i dettagli, vedere [SetRepositoryPolicy](#) in AWS CLI Command Reference.

## start-image-scan

Il seguente esempio di codice mostra come utilizzare `start-image-scan`.

### AWS CLI

Per avviare una scansione delle vulnerabilità di un'immagine

L'`start-image-scan` esempio seguente avvia una scansione di immagini per e viene specificata dall'`imageDigest` nel repository specificato.

```

aws ecr start-image-scan \
  --repository-name sample-repo \
  --image-
id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6

```

**Output:**

```
{
  "registryId": "012345678910",
  "repositoryName": "sample-repo",
  "imageId": {
    "imageDigest":
"sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
  },
  "imageScanStatus": {
    "status": "IN_PROGRESS"
  }
}
```

Per ulteriori informazioni, consulta [Image Scanning](#) nella Amazon ECR User Guide.

- Per API i dettagli, consulta [StartImageScan AWS CLI Command Reference](#).

**start-lifecycle-policy-preview**

Il seguente esempio di codice mostra come utilizzare `start-lifecycle-policy-preview`.

**AWS CLI**

Per creare un'anteprima della politica del ciclo di vita

L'`start-lifecycle-policy-preview` esempio seguente crea un'anteprima della politica del ciclo di vita definita da un JSON file per il repository specificato.

```
aws ecr start-lifecycle-policy-preview \
  --repository-name "project-a/amazon-ecs-sample" \
  --lifecycle-policy-text "file://policy.json"
```

Contenuto di `policy.json`.

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Expire images older than 14 days",
      "selection": {
        "tagStatus": "untagged",
```

```

        "countType": "sinceImagePushed",
        "countUnit": "days",
        "countNumber": 14
    },
    "action": {
        "type": "expire"
    }
}
]
}

```

### Output:

```

{
  "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\n\": \"days\",\n        \"countNumber\": 14\n      },\n      \"action\": {\n        \"type\": \"expire\"\n      }\n    }\n  ]\n}\n",
  "status": "IN_PROGRESS"
}

```

- Per API i dettagli, vedere [StartLifecyclePolicyPreview](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare un repository

L'esempio seguente imposta un tag con chiave `Stage` e valore `Integ` sul `hello-world` repository.

```

aws ecr tag-resource \
  --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \
  --tags Key=Stage,Value=Integ

```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da un repository

L'`untag-resource` esempio seguente rimuove il tag con la chiave `Stage` dal repository `hello-world`

```
aws ecr untag-resource \  
  --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \  
  --tag-keys Stage
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## upload-layer-part

Il seguente esempio di codice mostra come utilizzare `upload-layer-part`.

### AWS CLI

Per caricare una parte di livello

Di seguito viene `upload-layer-part` caricata una parte del livello di immagine nel `layer-test` repository.

```
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
```

Output:

```
{
  "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",
  "registryId": "012345678910",
  "lastByteReceived": 8323314,
  "repositoryName": "layer-test"
}
```

- Per API i dettagli, vedere [UploadLayerPart](#) in AWS CLI Command Reference.

## Esempi di Amazon ECR Public che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon ECR Public. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **create-repository**

Il seguente esempio di codice mostra come utilizzare `create-repository`.

#### AWS CLI

Esempio 1: creare un archivio in un registro pubblico

L'`create-repository` esempio seguente crea un repository denominato `project-a/nginx-web-app` in un registro pubblico.

```
aws ecr-public create-repository \
  --repository-name project-a/nginx-web-app
```

**Output:**

```
{
  "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": {}
}
```

Per ulteriori informazioni, consulta [Creating a public repository](#) nella Amazon ECR Public User Guide.

Esempio 2: creare un repository in un registro pubblico con una breve descrizione del contenuto del repository, del sistema e dell'architettura operativa con cui le immagini nel repository sono compatibili

L'create-repositoryesempio seguente crea un archivio denominato project-a/nginx-web-app in un registro pubblico con una breve descrizione del contenuto del repository, del sistema e dell'architettura operativa con cui le immagini del repository sono compatibili.

```
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'
```

**Output:**

```
{
  "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"
    ]
  }
}
```

Per ulteriori informazioni, consulta [Creating a public repository](#) nella Amazon ECR Public User Guide.

Esempio 3: creare un repository in un registro pubblico logolmageBlobaboutText, insieme a informazioni usageText e taggare

L'create-repositoryesempio seguente crea un archivio denominato project-a/ nginx-web-app in un registro pubblico logolmageBlob, aboutText insieme a e contrassegna le informazioni. usageText

```
aws ecr-public create-repository \
  --cli-input-json file://myfile.json
```

Contenuto di 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/03d54f8c4d522bf712cffd6c8f9aafba0a875e78/Dockerfile)\n* [`.2.0.20200722.0-
  with-sources`, `.2-with-sources`, `with-sources`](https://github.com/
  amazonlinux/container-images/blob/1e7349845e029a2e6afe6dc473ef17d052e3546f/
  Dockerfile)\n* [`.2018.03.0.20200602.1`, `2018.03`, `.1`](https://github.com/
  amazonlinux/container-images/blob/f10932e08c75457eeb372bf1cc47ea2a4b8e98c8/
  Dockerfile)\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"
    }
  ]
}

```

### Output:

```

{
  "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/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."
  }
}
```

Per ulteriori informazioni, consulta [Creazione di un repository pubblico](#) nella Amazon ECR Public User Guide e [i dati del catalogo Repository](#) nella Amazon ECR Public User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateRepositoryReference](#).

## delete-repository

Il seguente esempio di codice mostra come utilizzare `delete-repository`.

### AWS CLI

Per eliminare un repository in un registro pubblico

L'esempio seguente elimina un repository denominato `project-a/nginx-web-app` dal registro pubblico.

```
aws ecr-public delete-repository \  
  --repository-name project-a/nginx-web-app
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Eliminazione di un repository pubblico](#) nella Amazon ECR Public User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteRepositoryReference](#).

## ECSEsempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con Amazon ECS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-capacity-provider**

Il seguente esempio di codice mostra come utilizzare `create-capacity-provider`.

AWS CLI

Per creare un provider di capacità

L' `create-capacity-provider` seguente crea un provider di capacità che utilizza un gruppo Auto Scaling denominato MyASG, ha la scalabilità gestita e la protezione gestita dalla terminazione abilitata. Questa configurazione viene utilizzata per la scalabilità automatica dei ECS cluster Amazon.

```
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"
```

Output:

```
{
  "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": []
}

```

Per ulteriori informazioni, consulta [Amazon ECS cluster auto scaling](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [CreateCapacityProvider AWS CLI Command Reference](#).

## create-cluster

Il seguente esempio di codice mostra come utilizzare `create-cluster`.

### AWS CLI

Esempio 1: per creare un nuovo cluster

L'`create-cluster` seguente crea un cluster.

```

aws ecs create-cluster \
  --cluster-name MyCluster

```

Output:

```

{
  "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": []
  }
}

```

Per ulteriori informazioni, consulta [Creating a Cluster](#) nella Amazon ECS Developer Guide.

## Esempio 2: creare un nuovo cluster utilizzando fornitori di capacità

L'`create-cluster` seguente crea un cluster e vi associa due fornitori di capacità esistenti. Il `create-capacity-provider` comando viene utilizzato per creare un provider di capacità. La specificazione di una strategia predefinita per un provider di capacità è facoltativa, ma consigliata. In questo esempio, creiamo un cluster denominato `MyCluster` e ad esso associamo i `MyCapacityProvider1` fornitori di `MyCapacityProvider2` capacità. Viene specificata una strategia predefinita per i fornitori di capacità che distribuisce le attività in modo uniforme tra entrambi i fornitori di capacità.

```

aws ecs create-cluster --cluster-name MyCluster --capacity-providers MyCapacityProvider 1 2 --
= 1, peso=1 = 2, peso=1 MyCapacityProvider default-capacity-provider-strategy capacityProvider
MyCapacityProvider capacityProvider MyCapacityProvider

```

Output:

```

{
  "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": "ECManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
                }
            ]
        },
        {
            "id": "ae592060-2382-4663-9476-b015c685593c",
            "type": "asp",
            "status": "PRECREATED",
            "details": [
                {
                    "name": "capacityProviderName",
                    "value": "MyCapacityProvider2"
                },
                {
                    "name": "scalingPlanName",
                    "value": "ECManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222"
                }
            ]
        }
    ]
}

```

```

    }
  ]
}
  ],
  "attachmentsStatus": "UPDATE_IN_PROGRESS"
}
}

```

Per ulteriori informazioni, consulta [Cluster Capacity Provider](#) nella Amazon ECS Developer Guide.

Esempio 3: per creare un nuovo cluster con più tag

L'create-clustereseempio seguente crea un cluster con più tag. Per ulteriori informazioni sull'aggiunta di tag utilizzando la sintassi abbreviata, vedete Using Synthand [Syntax with the AWS Command Line Interface nella Guida per l'utente.AWS CLI](#)

```

aws ecs create-cluster \
  --cluster-name MyCluster \
  --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3

```

Output:

```

{
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Creating a Cluster](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [CreateCluster AWS CLI Command Reference](#).

## create-service

Il seguente esempio di codice mostra come utilizzare `create-service`.

### AWS CLI

Esempio 1: creare un servizio con un'attività Fargate

L'esempio seguente mostra come creare un servizio utilizzando un'attività Fargate.

```

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

```

Output:

```

{
  "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"
  }
}
```

Esempio 2: creare un servizio utilizzando il tipo di EC2 avvio

L'create-serviceesempio seguente mostra come creare un servizio chiamato ecs-simple-service con un'attività che utilizza il tipo di EC2 avvio. Il servizio utilizza la definizione dell'sleep360attività e mantiene 1 istanza dell'attività.

```
aws ecs create-service \
  --cluster MyCluster \
  --service-name ecs-simple-service \
  --task-definition sleep360:2 \
  --desired-count 1
```

## Output:

```
{
  "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"
  }
}
```

```
}
```

Esempio 3: creare un servizio che utilizza un controller di distribuzione esterno

L'create-serviceesempio seguente crea un servizio che utilizza un controller di distribuzione esterno.

```
aws ecs create-service \  
  --cluster MyCluster \  
  --service-name MyService \  
  --deployment-controller type=EXTERNAL \  
  --desired-count 1
```

Output:

```
{  
  "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"  
    }  
  }  
}
```



```

    },
    "enableECSManagedTags": false,
    "propagateTags": "NONE"
  }
}

```

#### Esempio 4: creare un nuovo servizio basato su un sistema di bilanciamento del carico

L'`create-service` seguente mostra come creare un servizio basato su un sistema di bilanciamento del carico. È necessario disporre di un sistema di bilanciamento del carico configurato nella stessa regione dell'istanza del contenitore. Questo esempio utilizza l'`--cli-input-json` opzione e un file JSON di input chiamato `ecs-simple-service-elb.json` con il seguente contenuto:

```

{
  "serviceName": "ecs-simple-service-elb",
  "taskDefinition": "ecs-demo",
  "loadBalancers": [
    {
      "loadBalancerName": "EC2Contai-EcsElast-123456789012",
      "containerName": "simple-demo",
      "containerPort": 80
    }
  ],
  "desiredCount": 10,
  "role": "ecsServiceRole"
}

```

Comando:

```

aws ecs create-service \
  --cluster MyCluster \
  --service-name ecs-simple-service-elb \
  --cli-input-json file://ecs-simple-service-elb.json

```

Output:

```

{
  "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
}
}
```

Per ulteriori informazioni, consulta [Creating a Service](#) nell'Amazon ECS Developer Guide.

- Per API i dettagli, consulta [CreateService AWS CLI Command Reference](#).

## create-task-set

Il seguente esempio di codice mostra come utilizzare `create-task-set`.

### AWS CLI

Per creare un set di attività

L'create-task-setesempio seguente crea un set di attività in un servizio che utilizza un controller di distribuzione esterno.

```
aws ecs create-task-set \  
  --cluster MyCluster \  
  --service MyService \  
  --task-definition MyTaskDefinition:2 \  
  --network-  
configuration "awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321]}"
```

Output:

```
{  
  "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
  }
}
```

- Per API i dettagli, vedere [CreateTaskSet](#) in AWS CLI Command Reference.

## delete-account-setting

Il seguente esempio di codice mostra come utilizzare `delete-account-setting`.

### AWS CLI

Per eliminare le impostazioni dell'account per un IAM utente o un IAM ruolo specifico

L'esempio seguente `delete-account-setting` elimina le impostazioni dell'account per l'IAM utente o il IAM ruolo specifico.

```
aws ecs delete-account-setting \
  --name serviceLongArnFormat \
  --principal-arn arn:aws:iam::123456789012:user/MyUser
```

Output:

```
{
  "setting": {
    "name": "serviceLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::123456789012:user/MyUser"
  }
}
```

Per ulteriori informazioni, consulta [Amazon Resource Names \(ARNs\) e IDs](#) l'Amazon ECS Developer Guide.

- Per API i dettagli, consulta [DeleteAccountSetting AWS CLI](#) Command Reference.

## delete-attributes

Il seguente esempio di codice mostra come utilizzare `delete-attributes`.

## AWS CLI

Per eliminare uno o più attributi personalizzati da una ECS risorsa Amazon

Quanto segue `delete-attributes` elimina un attributo con lo stesso nome `stack` da un'istanza del contenitore.

```
aws ecs delete-attributes \  
  --attributes name=stack, targetId=arn:aws:ecs:us-west-2:130757420319:container-  
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

Output:

```
{  
  "attributes": [  
    {  
      "name": "stack",  
      "targetId": "arn:aws:ecs:us-west-2:130757420319:container-  
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",  
      "value": "production"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [DeleteAttributes](#) in AWS CLI Command Reference.

## **delete-capacity-provider**

Il seguente esempio di codice mostra come utilizzare `delete-capacity-provider`.

### AWS CLI

Esempio 1: eliminare un provider di capacità utilizzando Amazon Resource Name (ARN)

L'`delete-capacity-provider` esempio seguente elimina un provider di capacità specificando l'Amazon Resource Name (ARN) del fornitore di capacità. Lo stato e ARN lo stato dell'eliminazione del provider di capacità possono essere recuperati utilizzando il comando `describe-capacity-providers`

```
aws ecs delete-capacity-provider \  
  --arn
```

```
--capacity-provider arn:aws:ecs:us-west-2:123456789012:capacity-provider/ExampleCapacityProvider
```

Output:

```
{
  "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": []
  }
}
```

Per ulteriori informazioni, consulta [Cluster Capacity Provider](#) nella Amazon ECS Developer Guide.

Esempio 2: eliminare un provider di capacità utilizzando il nome

L'`delete-capacity-provider` seguente elimina un fornitore di capacità specificando il nome breve del fornitore di capacità. Il nome breve e lo stato dell'eliminazione del provider di capacità possono essere recuperati utilizzando il comando `describe-capacity-providers`

```
aws ecs delete-capacity-provider \  
--capacity-provider ExampleCapacityProvider
```

Output:

```
{
```

```

    "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": []
    }
  }
}

```

Per ulteriori informazioni, consulta [Cluster Capacity Provider](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [DeleteCapacityProvider AWS CLI Command Reference](#).

## delete-cluster

Il seguente esempio di codice mostra come utilizzare `delete-cluster`.

### AWS CLI

Per eliminare un cluster vuoto

L'`delete-cluster` esempio seguente elimina il cluster vuoto specificato.

```
aws ecs delete-cluster --cluster MyCluster
```

Output:

```

{
  "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": []
  }
}
```

Per ulteriori informazioni, consulta la sezione [Eliminazione di un cluster](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [DeleteCluster AWS CLI Command Reference](#).

## delete-service

Il seguente esempio di codice mostra come utilizzare `delete-service`.

### AWS CLI

Per eliminare un servizio

L'ecs `delete-service` esempio seguente elimina il servizio specificato da un cluster. È possibile includere il `--force` parametro per eliminare un servizio anche se non è stato ridimensionato a zero attività.

```
aws ecs delete-service --cluster MyCluster --service MyService1 --force
```

Per ulteriori informazioni, consulta [Eliminazione di un servizio](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [DeleteService AWS CLI Command Reference](#).

## delete-task-definitions

Il seguente esempio di codice mostra come utilizzare `delete-task-definitions`.

### AWS CLI

Per eliminare una definizione di attività



L'`delete-task-definition` seguente elimina una definizione di INACTIVE attività.

```
aws ecs delete-task-definitions \  
  --task-definition curltest:1
```

Output:

```
{  
  "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": []
}

```

Per ulteriori informazioni, consulta [Amazon ECS Task Definitions](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [DeleteTaskDefinitions AWS CLI Command Reference](#).

## delete-task-set

Il seguente esempio di codice mostra come utilizzare `delete-task-set`.

### AWS CLI

Per eliminare un set di attività

L'`delete-task-set` esempio seguente mostra come eliminare un set di attività. È possibile includere il `--force` parametro per eliminare un set di attività anche se non è stato ridimensionato a zero.

```

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
```

Output:

```
{
  "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
  }
}
```

- Per API i dettagli, vedere [DeleteTaskSet](#) in AWS CLI Command Reference.

## deregister-container-instance

Il seguente esempio di codice mostra come utilizzare `deregister-container-instance`.

### AWS CLI

Per annullare la registrazione di un'istanza di contenitore da un cluster

L'`deregister-container-instance` esempio seguente annulla la registrazione di un'istanza di contenitore dal cluster specificato. Se ci sono ancora attività in esecuzione nell'istanza del contenitore, è necessario interromperle prima di annullare la registrazione oppure utilizzare l'opzione `--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
```

Output:

```
{
  "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"
}
}

```

Per ulteriori informazioni, consulta [Annullare la registrazione di un'istanza del contenitore](#) nella Guida per gli sviluppatori. ECS

- Per API i dettagli, consulta AWS CLI Command [DeregisterContainerInstance](#) Reference.

## deregister-task-definition

Il seguente esempio di codice mostra come utilizzare `deregister-task-definition`.

### AWS CLI

Per annullare la registrazione di una definizione di attività

L'esempio seguente annulla la registrazione della prima revisione della definizione dell'attività `curler` nell'area predefinita.

```
aws ecs deregister-task-definition --task-definition curler:1
```

Si noti che nell'output risultante, lo stato della definizione dell'attività mostra: 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
}
}
```

Per ulteriori informazioni, consulta [Amazon ECS Task Definitions](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [DeregisterTaskDefinition AWS CLI Command Reference](#).

## describe-capacity-providers

Il seguente esempio di codice mostra come utilizzare `describe-capacity-providers`.

### AWS CLI

Esempio 1: Per descrivere tutti i fornitori di capacità

L'`describe-capacity-providers` seguente recupera i dettagli su tutti i fornitori di capacità.

```
aws ecs describe-capacity-providers
```

Output:

```
{
  "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": []
}
]
}

```

Per ulteriori informazioni, consulta [Cluster Capacity Provider](#) nella Amazon ECS Developer Guide.

Esempio 2: per descrivere uno specifico fornitore di capacità

L'`describe-capacity-providers` seguente recupera i dettagli su uno specifico fornitore di capacità. L'utilizzo del `--include TAGS` parametro aggiungerà all'output i tag associati al fornitore di capacità.

```

aws ecs describe-capacity-providers \
  --capacity-providers MyCapacityProvider \
  --include TAGS

```

Output:

```
{
  "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"
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, consulta [Cluster Capacity Provider](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [DescribeCapacityProviders AWS CLI Command Reference](#).

## describe-clusters

Il seguente esempio di codice mostra come utilizzare `describe-clusters`.

### AWS CLI

Esempio 1: Per descrivere un cluster

L'`describe-clusters` seguente recupera i dettagli sul cluster specificato.

```
aws ecs describe-clusters \  
  --cluster default
```

Output:

```
{  
  "clusters": [  
    {  
      "status": "ACTIVE",  
      "clusterName": "default",  
      "registeredContainerInstancesCount": 0,  
      "pendingTasksCount": 0,  
      "runningTasksCount": 0,  
      "activeServicesCount": 1,  
      "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default"  
    }  
  ],  
  "failures": []  
}
```

Per ulteriori informazioni, consulta [Amazon ECS Clusters](#) nell'Amazon ECS Developer Guide.

Esempio 2: descrivere un cluster con l'opzione attachment

L'opzione `describe-clusters` seguente specifica l'opzione `ATTACHMENTS`. Recupera i dettagli sul cluster specificato e un elenco di risorse collegate al cluster sotto forma di allegati. Quando si utilizza un provider di capacità con un cluster, le risorse, sia che si tratti di AutoScaling piani o politiche di scalabilità, saranno rappresentate come `asp` o `as_policy`. `ATTACHMENTS`

```
aws ecs describe-clusters \  
  --include ATTACHMENTS \  
  --clusters sampleCluster
```

Output:

```
{  
  "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": "ECManagedAutoScalingPolicy-3048e262-
fe39-4eaf-826d-6f975d303188"
          }
        ]
      }
    ],
    "attachmentsStatus": "UPDATE_COMPLETE"
  },
  "failures": []
}

```

Per ulteriori informazioni, consulta [Amazon ECS Clusters](#) nell'Amazon ECS Developer Guide.

- Per API i dettagli, consulta [DescribeClusters AWS CLI](#) Command Reference.

## describe-container-instances

Il seguente esempio di codice mostra come utilizzare `describe-container-instances`.

## AWS CLI

Per descrivere l'istanza del contenitore

L'`describe-container-instances` seguente recupera i dettagli di un'istanza di contenitore nel `update` cluster, utilizzando l'istanza del contenitore UUID come identificatore.

```
aws ecs describe-container-instances \
  --cluster update \
  --container-instances a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Output:

```
{
  "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"
    }
  }
}
```

```
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Amazon ECS Container Instances](#) nell'Amazon ECS Developer Guide.

- Per API i dettagli, consulta [DescribeContainerInstances AWS CLI Command Reference](#).

## describe-services

Il seguente esempio di codice mostra come utilizzare `describe-services`.

### AWS CLI

Per descrivere un servizio

L'`describe-services` esempio seguente recupera i dettagli del `my-http-service` servizio nel cluster predefinito.

```
aws ecs describe-services --services my-http-service
```

Output:

```
{  
  "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,  
          "loadBalancers": [],  
          "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/  
amazon-ecs-sample:1",  
          "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/my-http-  
service",  
          "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",  
          "taskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-definition/  
amazon-ecs-sample:1",  
          "deploymentConfiguration": {  
            "rollback": "REVERT_TO_PREVIOUS_DEPLOYMENT",  
            "strategy": "REPLACEMENT",  
            "type": "ECS_UPDATE_SERVICE"},  
          "deploymentOrder": 1,  
          "desiredCount": 10,  
          "pendingCount": 0,  
          "status": "PRIMARY",  
          "updatedAt": 1466801808.595,  
          "createdAt": 1466801808.595,  
          "id": "my-http-service-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": []
}

```

Per ulteriori informazioni, consulta la sezione [Servizi](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [DescribeServices AWS CLI Command Reference](#).

## describe-task-definition

Il seguente esempio di codice mostra come utilizzare `describe-task-definition`.

### AWS CLI

Per descrivere la definizione di un'attività

L'`describe-task-definition` esempio seguente recupera i dettagli di una definizione di attività.

```
aws ecs describe-task-definition \
  --task-definition hello_world:8
```

Output:

```
{
```

```
"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": []
}
```

Per ulteriori informazioni, consulta [Amazon ECS Task Definitions](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [DescribeTaskDefinition AWS CLI Command Reference](#).

## describe-task-sets

Il seguente esempio di codice mostra come utilizzare `describe-task-sets`.

### AWS CLI

Per descrivere un set di attività

L'esempio seguente descrive un set di attività in un servizio che utilizza un deployer esterno.

```
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
```

Output:

```
{
  "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": []
}
```

- Per API i dettagli, vedere [DescribeTaskSets](#) in AWS CLI Command Reference.

## describe-tasks

Il seguente esempio di codice mostra come utilizzare `describe-tasks`.

### AWS CLI

Esempio 1: per descrivere una singola attività

L'`describe-tasks` seguente recupera i dettagli di un'attività in un cluster. È possibile specificare l'attività utilizzando l'ID o il valore full ARN dell'attività. In questo esempio viene utilizzato il ARN valore completo del task.

```
aws ecs describe-tasks \  
  --cluster MyCluster \  
  --tasks arn:aws:ecs:us-east-1:123456789012:task/  
MyCluster/4d590253bb114126b7afa7b58EXAMPLE
```

Output:

```
{  
  "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": []
}

```

Per ulteriori informazioni, consulta [Amazon ECS Task Definitions](#) nella Amazon ECS Developer Guide.



## Esempio 2: per descrivere più attività

L' `describe-tasks` seguente recupera i dettagli di più attività in un cluster. È possibile specificare l'attività utilizzando l'ID o il valore full ARN dell'attività. In questo esempio viene utilizzata la totalità IDs delle attività.

```
aws ecs describe-tasks \  
  --cluster MyCluster \  
  --tasks "74de0355a10a4f979ac495c14EXAMPLE" "d789e94343414c25b9f6bd59eEXAMPLE"
```

Output:

```
{  
  "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": []
}
```

Per ulteriori informazioni, consulta [Amazon ECS Task Definitions](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [DescribeTasks AWS CLI Command Reference](#).

## execute-command

Il seguente esempio di codice mostra come utilizzare `execute-command`.



```

    {
      "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"
    }
  ]
}

```

Esempio 2: Per visualizzare le impostazioni dell'account per un IAM utente o un IAM ruolo specifico

L'`list-account-settings` seguente visualizza le impostazioni dell'account per l'IAM utente o il IAM ruolo specificato.

```
aws ecs list-account-settings --principal-arn arn:aws:iam::123456789012:user/MyUser
```

Output:

```

{
  "settings": [
    {
      "name": "serviceLongArnFormat",
      "value": "enabled",
      "principalArn": "arn:aws:iam::123456789012:user/MyUser"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Amazon Resource Names \(ARNs\) e IDs](#) l'Amazon ECS Developer Guide.

- Per API i dettagli, consulta [ListAccountSettings AWS CLI Command Reference](#).

## list-attributes

Il seguente esempio di codice mostra come utilizzare `list-attributes`.

### AWS CLI

Per elencare le istanze del contenitore che contengono un attributo specifico

L'esempio seguente elenca gli attributi per le istanze di contenitore che hanno l'`stack=production` attributo nel cluster predefinito.

```
aws ecs list-attributes \
  --target-type container-instance \
  --attribute-name stack \
  --attribute-value production \
  --cluster default
```

Output:

```
{
  "attributes": [
    {
      "name": "stack",
      "targetId": "arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
      "value": "production"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Amazon ECS Container Agent Configuration](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [ListAttributes AWS CLI Command Reference](#).

## list-clusters

Il seguente esempio di codice mostra come utilizzare `list-clusters`.

### AWS CLI

Per elencare i cluster disponibili



L'`list-clusters` seguente elenca tutti i cluster disponibili.

```
aws ecs list-clusters
```

Output:

```
{
  "clusterArns": [
    "arn:aws:ecs:us-west-2:123456789012:cluster/MyECSCluster1",
    "arn:aws:ecs:us-west-2:123456789012:cluster/AnotherECSCluster"
  ]
}
```

Per ulteriori informazioni, consulta [Amazon ECS Clusters](#) nell'Amazon ECS Developer Guide.

- Per API i dettagli, consulta [ListClusters AWS CLI Command Reference](#).

## `list-container-instances`

Il seguente esempio di codice mostra come utilizzare `list-container-instances`.

AWS CLI

Per elencare le istanze del contenitore in un cluster

L'`list-container-instances` seguente elenca tutte le istanze di container disponibili in un cluster.

```
aws ecs list-container-instances --cluster MyCluster
```

Output:

```
{
  "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"
  ]
}
```

Per ulteriori informazioni, consulta [Amazon ECS Container Instances](#) nell'Amazon ECS Developer Guide.

- Per API i dettagli, consulta [ListContainerInstances AWS CLI Command Reference](#).

## **list-services-by-namespace**

Il seguente esempio di codice mostra come utilizzare `list-services-by-namespace`.

### AWS CLI

Per elencare i servizi in un namespace

L'`list-services-by-namespace` esempio seguente elenca tutti i servizi configurati per lo spazio dei nomi specificato nella regione predefinita.

```
aws ecs list-services-by-namespace \  
  --namespace service-connect
```

Output:

```
{  
  "serviceArns": [  
    "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService",  
    "arn:aws:ecs:us-west-2:123456789012:service/tutorial/service-connect-nginx-  
service"  
  ]  
}
```

Per ulteriori informazioni, consulta [Service Connect](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [ListServicesByNamespace AWS CLI Command Reference](#).

## **list-services**

Il seguente esempio di codice mostra come utilizzare `list-services`.

### AWS CLI

Per elencare i servizi in un cluster

L'`list-services` esempio seguente mostra come elencare i servizi in esecuzione in un cluster.

```
aws ecs list-services --cluster MyCluster
```

Output:

```
{
  "serviceArns": [
    "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService"
  ]
}
```

Per ulteriori informazioni, consulta la sezione [Servizi](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [ListServices AWS CLI Command Reference](#).

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

AWS CLI

Per elencare i tag di una risorsa

L'`list-tags-for-resource` esempio seguente elenca i tag per un cluster specifico.

```
aws ecs list-tags-for-resource \
  --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster
```

Output:

```
{
  "tags": [
    {
      "key": "key1",
      "value": "value1"
    },
    {
      "key": "key2",
      "value": "value2"
    }
  ]
}
```

```
    },
    {
      "key": "key3",
      "value": "value3"
    }
  ]
}
```

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## list-task-definition-families

Il seguente esempio di codice mostra come utilizzare `list-task-definition-families`.

### AWS CLI

Esempio 1: per elencare le famiglie di definizioni delle attività registrate

L'`list-task-definition-families` esempio seguente elenca tutte le famiglie di definizioni di attività registrate.

```
aws ecs list-task-definition-families
```

Output:

```
{
  "families": [
    "node-js-app",
    "web-timer",
    "hpcc",
    "hpcc-c4-8xlarge"
  ]
}
```

Esempio 2: per filtrare le famiglie di definizioni delle attività registrate

L'`list-task-definition-families` esempio seguente elenca le revisioni delle definizioni delle attività che iniziano con «hpcc».

```
aws ecs list-task-definition-families --family-prefix hpcc
```

Output:

```
{
  "families": [
    "hpc",
    "hpc-c4-8xlarge"
  ]
}
```

Per ulteriori informazioni, consulta [Task Definition Parameters](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [ListTaskDefinitionFamilies AWS CLI Command Reference](#).

## list-task-definitions

Il seguente esempio di codice mostra come utilizzare `list-task-definitions`.

AWS CLI

Esempio 1: per elencare le definizioni delle attività registrate

L'`list-task-definitions` seguente elenca tutte le definizioni delle attività registrate.

```
aws ecs list-task-definitions
```

Output:

```
{
  "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"
  ]
}
```

Esempio 2: Elencare le definizioni delle attività registrate in una famiglia

L' `list-task-definitions` esempio seguente elenca le revisioni delle definizioni delle attività di una famiglia specificata.

```
aws ecs list-task-definitions --family-prefix wordpress
```

Output:

```
{
  "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"
  ]
}
```

Per ulteriori informazioni, consulta [Amazon ECS Task Definitions](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [ListTaskDefinitions AWS CLI Command Reference](#).

## list-tasks

Il seguente esempio di codice mostra come utilizzare `list-tasks`.

AWS CLI

Esempio 1: per elencare le attività in un cluster

L' `list-tasks` esempio seguente elenca tutte le attività di un cluster.

```
aws ecs list-tasks --cluster default
```

Output:

```
{
  "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"
  ]
}
```

```
]
}
```

Esempio 2: per elencare le attività su una particolare istanza del contenitore

L'`list-tasks` seguente elenca le attività su un'istanza del contenitore, utilizzando l'istanza del contenitore UUID come filtro.

```
aws ecs list-tasks --cluster default --container-instance a1b2c3d4-5678-90ab-cdef-3333EXAMPLE
```

Output:

```
{
  "taskArns": [
    "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
  ]
}
```

Per ulteriori informazioni, consulta [Amazon ECS Task Definitions](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [ListTasks AWS CLI Command Reference](#).

## `put-account-setting-default`

Il seguente esempio di codice mostra come utilizzare `put-account-setting-default`.

AWS CLI

Per modificare le impostazioni predefinite dell'account

L'`put-account-setting-default` seguente modifica l'impostazione predefinita dell'account per tutti IAM gli utenti o i ruoli dell'account. Queste modifiche si applicano all'intero AWS account a meno che un IAM utente o un ruolo non sostituisca esplicitamente queste impostazioni.

```
aws ecs put-account-setting-default --name serviceLongArnFormat --value enabled
```

**Output:**

```
{
  "setting": {
    "name": "serviceLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::123456789012:root"
  }
}
```

Per ulteriori informazioni, consulta [Amazon Resource Names \(ARNs\) e IDs](#) l'Amazon ECS Developer Guide.

- Per API i dettagli, consulta [PutAccountSettingDefault AWS CLI](#) Command Reference.

**put-account-setting**

Il seguente esempio di codice mostra come utilizzare `put-account-setting`.

**AWS CLI**

Per modificare le impostazioni IAM dell'account utente

L'`put-account-setting` esempio seguente abilita l'`serviceLongArnFormat` impostazione IAM dell'account utente.

```
aws ecs put-account-setting --name serviceLongArnFormat --value enabled
```

**Output:**

```
{
  "setting": {
    "name": "serviceLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::130757420319:user/your_username"
  }
}
```

Per ulteriori informazioni, consulta [Modifica delle impostazioni dell'account](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [PutAccountSetting AWS CLI](#) Command Reference.



## put-account-settings

Il seguente esempio di codice mostra come utilizzare `put-account-settings`.

### AWS CLI

Per modificare le impostazioni dell'account per un IAM utente o un IAM ruolo

L'esempio seguente modifica le impostazioni dell'account per l'IAM utente o il IAM ruolo specificato.

```
aws ecs put-account-setting \  
  --name serviceLongArnFormat \  
  --value enabled \  
  --principal-arn arn:aws:iam::123456789012:user/MyUser
```

Output:

```
{  
  "setting": {  
    "name": "serviceLongArnFormat",  
    "value": "enabled",  
    "principalArn": "arn:aws:iam::123456789012:user/MyUser"  
  }  
}
```

- Per API i dettagli, vedere [PutAccountSettings](#) in AWS CLI Command Reference.

## put-attributes

Il seguente esempio di codice mostra come utilizzare `put-attributes`.

### AWS CLI

Per creare un attributo e associarlo a una ECS risorsa Amazon

Quanto segue `put-attributes` applica un attributo con il nome `stack` e il valore `production` a un'istanza di contenitore.

```
aws ecs put-attributes \  
  --attributes name=stack,value=production,targetId=arn:aws:ecs:us-west-2:130757420319:container-instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

**Output:**

```
{
  "attributes": [
    {
      "name": "stack",
      "targetId": "arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
      "value": "production"
    }
  ]
}
```

- Per API i dettagli, vedere [PutAttributes](#) in AWS CLI Command Reference.

**put-cluster-capacity-providers**

Il seguente esempio di codice mostra come utilizzare `put-cluster-capacity-providers`.

**AWS CLI**

Esempio 1: aggiungere un provider di capacità esistente a un cluster

L'`put-cluster-capacity-providers` esempio seguente aggiunge un provider di capacità esistente a un cluster. Il `create-capacity-provider` comando viene utilizzato per creare un provider di capacità. Il `describe-clusters` comando viene utilizzato per descrivere gli attuali fornitori di capacità e la strategia predefinita dei fornitori di capacità associata a un cluster. Quando si aggiunge un nuovo fornitore di capacità a un cluster, è necessario specificare tutti i fornitori di capacità esistenti oltre al nuovo fornitore di capacità che si desidera associare al cluster. È inoltre necessario specificare la strategia predefinita del provider di capacità da associare al cluster. In questo esempio, al `MyCluster` cluster è associato il provider di `MyCapacityProvider1` capacità e si desidera aggiungere il fornitore di `MyCapacityProvider2` capacità e includerlo nella strategia predefinita del fornitore di capacità in modo che le attività siano distribuite uniformemente tra entrambi i fornitori di capacità.

```
aws ecs put-cluster-capacity-providers \
  --cluster MyCluster \
  --capacity-providers MyCapacityProvider1 MyCapacityProvider2 \
  --default-capacity-provider-
strategy capacityProvider=MyCapacityProvider1,weight=1 capacityProvider=MyCapacityProvider2,
```

## Output:

```
{
  "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": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
        }
    ]
},
{
    "id": "ae592060-2382-4663-9476-b015c685593c",
    "type": "as_policy",
    "status": "ACTIVE",
    "details": [
        {
            "name": "capacityProviderName",
            "value": "MyCapacityProvider2"
        },
        {
            "name": "scalingPolicyName",
            "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
        }
    ]
}
],
"attachmentsStatus": "UPDATE_IN_PROGRESS"
}
}

```

Per ulteriori informazioni, consulta [Cluster Capacity Provider](#) nella Amazon ECS Developer Guide.

### Esempio 2: rimuovere un fornitore di capacità da un cluster

L'put-cluster-capacity-providersesempio seguente rimuove un fornitore di capacità da un cluster. Il describe-clusters comando viene utilizzato per descrivere gli attuali provider di capacità associati a un cluster. Quando si rimuove un fornitore di capacità da un cluster, è necessario specificare i fornitori di capacità che si desidera che rimangano associati al cluster, nonché la strategia predefinita del provider di capacità da associare al cluster. In questo esempio, al cluster sono associati i provider di MyCapacityProvider2 capacità MyCapacityProvider1 e si desidera rimuovere il provider di MyCapacityProvider2 capacità, quindi lo si specifica solo MyCapacityProvider1 nel comando insieme alla strategia predefinita aggiornata del provider di capacità.

```
aws ecs put-cluster-capacity-providers \  
  --cluster MyCluster \  
  --capacity-providers MyCapacityProvider1 \  
  --default-capacity-provider-  
strategy capacityProvider=MyCapacityProvider1,weight=1,base=0
```

Output:

```
{  
  "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"
}
}

```

Per ulteriori informazioni, consulta [Cluster Capacity Provider](#) nella Amazon ECS Developer Guide.

Esempio 3: rimuovere tutti i provider di capacità da un cluster

L'`put-cluster-capacity-providers` seguente rimuove tutti i provider di capacità esistenti dal cluster.

```

aws ecs put-cluster-capacity-providers \
  --cluster MyCluster \
  --capacity-providers [] \
  --default-capacity-provider-strategy []

```

Output:

```
{
```

```
"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": "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"
}
}

```

Per ulteriori informazioni, consulta [Cluster Capacity Provider](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [PutClusterCapacityProviders AWS CLI Command Reference](#).

## register-task-definition

Il seguente esempio di codice mostra come utilizzare `register-task-definition`.

### AWS CLI

Esempio 1: registrare una definizione di attività con un JSON file

L'esempio seguente registra una definizione di attività nella famiglia specificata. Le definizioni dei contenitori vengono salvate in JSON formato nella posizione del file specificata.

```
aws ecs register-task-definition \
  --cli-input-json file://<path_to_json_file>/sleep360.json
```

Contenuto di `sleep360.json`.

```

{
  "containerDefinitions": [
    {
      "name": "sleep",
      "image": "busybox",
      "cpu": 10,
      "command": [
        "sleep",

```



```
        "360"
      ],
      "memory": 10,
      "essential": true
    }
  ],
  "family": "sleep360"
}
```

### Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [le definizioni di attività di esempio](#) nella Amazon ECS Developer Guide.

Esempio 2: per registrare una definizione di attività con un parametro JSON stringa

L'`register-task-definition`esempio seguente registra una definizione di attività utilizzando le definizioni dei contenitori fornite come parametro di JSON stringa con virgolette doppie con escape.

```
aws ecs register-task-definition \  
  --family sleep360 \  
  --container-definitions "[{\"name\": \"sleep\", \"image\": \"busybox\", \"cpu\": 10, \"command\": [\"sleep\", \"360\"], \"memory\": 10, \"essential\": true}]"
```

L'output è identico all'esempio precedente.

Per ulteriori informazioni, consulta [Creating a Task Definition](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [RegisterTaskDefinition AWS CLI Command Reference](#).

## run-task

Il seguente esempio di codice mostra come utilizzare `run-task`.

### AWS CLI

Per eseguire un'attività nel cluster predefinito

L'`run-task`esempio seguente esegue un'attività nel cluster predefinito e utilizza un token client.

```
aws ecs run-task \  
  --cluster default \  
  --task-definition sleep360:1 \  
  --client-token 550e8400-e29b-41d4-a716-446655440000
```

Output:

```
{  
  "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": []
}

```

Per ulteriori informazioni, consulta [Running Tasks](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [RunTask AWS CLI Command Reference](#).

## start-task

Il seguente esempio di codice mostra come utilizzare `start-task`.

### AWS CLI

Per iniziare una nuova attività

Quanto segue `start-task` avvia un'attività utilizzando la revisione più recente della definizione dell'`sleep360` attività sull'istanza del contenitore specificata nel cluster predefinito.

```

aws ecs start-task \
  --task-definition sleep360 \
  --container-instances 765936fadbdd46b5991a4bd70c2a43d4

```

Output:

```

{
  "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/666fdccc2e2d4b6894dd422f4e8f8",
      "name": "sleep",
      "lastStatus": "PENDING",
      "networkInterfaces": [],
      "cpu": "10",
      "memory": "10"
    }
  ],
  "version": 1,
  "createdAt": 1563421494.186,
  "group": "family:sleep360",
  "launchType": "EC2",
  "attachments": [],
  "tags": []
}
],
"failures": []
}

```

- Per API i dettagli, vedere [StartTask](#) in AWS CLI Command Reference.

## stop-task

Il seguente esempio di codice mostra come utilizzare `stop-task`.

### AWS CLI

Per interrompere un'operazione

Quanto segue `stop-task` interrompe l'esecuzione dell'attività specificata nel cluster predefinito.

```
aws ecs stop-task \  
--task 666fdccc2e2d4b6894dd422f4eeee8f8
```

Output:

```
{  
  "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": []  
  }  
}
```

- Per API i dettagli, vedere [StopTask](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare una risorsa

L'`tag-resource` esempio seguente aggiunge un singolo tag alla risorsa specificata.

```
aws ecs tag-resource \  
  --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \  
  --tags key=key1,value=value1
```

Questo comando non produce alcun output.

Per aggiungere più tag a una risorsa

L'`tag-resource` esempio seguente aggiunge più tag alla risorsa specificata.

```
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
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da una risorsa

L'`untag-resource` esempio seguente rimuove i tag elencati dalla risorsa specificata.

```
aws ecs untag-resource \  
  --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \  
  --tag-keys key1,key2
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-cluster-settings

Il seguente esempio di codice mostra come utilizzare `update-cluster-settings`.

### AWS CLI

Per modificare le impostazioni del cluster

L'esempio seguente abilita CloudWatch Container Insights per il default cluster.

```
aws ecs update-cluster-settings \  
  --cluster default \  
  --settings name=containerInsights,value=enabled
```

Output:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Modifica delle impostazioni dell'account](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [UpdateClusterSettings AWS CLI](#) Command Reference.



## update-container-agent

Il seguente esempio di codice mostra come utilizzare `update-container-agent`.

### AWS CLI

Per aggiornare l'agente container su un'istanza di Amazon ECS Container

L'`update-container-agent` seguente aggiorna l'agente del contenitore sull'istanza di contenitore specificata nel cluster predefinito.

```
aws ecs update-container-agent --cluster default --container-  
instance a1b2c3d4-5678-90ab-cdef-1111EXAMPLE
```

Output:

```
{  
  "containerInstance": {  
    "status": "ACTIVE",  
    ...  
    "agentUpdateStatus": "PENDING",  
    "versionInfo": {  
      "agentVersion": "1.0.0",  
      "agentHash": "4023248",  
      "dockerVersion": "DockerVersion: 1.5.0"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Updating the Amazon ECS Container Agent](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [UpdateContainerAgent AWS CLI Command Reference](#).

## update-container-instances-state

Il seguente esempio di codice mostra come utilizzare `update-container-instances-state`.

### AWS CLI

Per aggiornare lo stato di un'istanza del contenitore

Quanto segue `update-container-instances-state` aggiorna lo stato dell'istanza del contenitore specificata in `DRAINING` cui verrà rimossa dal cluster in cui è registrata.

```
aws ecs update-container-instances-state \  
  --container-instances 765936fadbdd46b5991a4bd70c2a43d4 \  
  --status DRAINING
```

Output:

```
{  
  "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": "91ccef8-2019.06.0"
    },
    {
      "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": []
}

```

- Per API i dettagli, vedere [UpdateContainerInstancesState](#) in AWS CLI Command Reference.

## update-service-primary-task-set

Il seguente esempio di codice mostra come utilizzare `update-service-primary-task-set`.

### AWS CLI

Per aggiornare il set di attività principale per un servizio

L'`update-service-primary-task-set` esempio seguente aggiorna il set di attività principale per il servizio specificato.

```

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

```

Output:

```

{
  "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
  }
}

```

- Per API i dettagli, vedere [UpdateServicePrimaryTaskSet](#) in AWS CLI Command Reference.

## update-service

Il seguente esempio di codice mostra come utilizzare `update-service`.

### AWS CLI

Esempio 1: Per modificare la definizione dell'attività utilizzata in un servizio

L'`update-service` esempio seguente aggiorna il `my-http-service` servizio per utilizzare la definizione dell'`amazon-ecs-sample` attività.

```
aws ecs update-service --service my-http-service --task-definition amazon-ecs-sample
```

Esempio 2: Per modificare il numero di attività in un servizio

L'update-serviceesempio seguente aggiorna il numero di attività desiderato del servizio my-http-service a 3.

```
aws ecs update-service --service my-http-service --desired-count 3
```

Per ulteriori informazioni, consulta la sezione [Aggiornamento di un servizio](#) nella Amazon ECS Developer Guide.

- Per API i dettagli, consulta [UpdateService AWS CLI](#) Command Reference.

## update-task-set

Il seguente esempio di codice mostra come utilizzareupdate-task-set.

### AWS CLI

Per aggiornare un set di attività

L'update-task-setesempio seguente aggiorna un set di attività per regolare la scala.

```
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
```

Output:

```
{  
  "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  
  }  
}
```

- Per API i dettagli, vedere [UpdateTaskSet](#) in AWS CLI Command Reference.

## EFSEsempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con AmazonEFS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **create-file-system**

Il seguente esempio di codice mostra come utilizzare `create-file-system`.

#### AWS CLI

Per creare un file system crittografato

L'esempio seguente crea un file system crittografato utilizzando l'impostazione predefinita CMK. Aggiunge anche il `tagName=my-file-system`.

```
aws efs create-file-system \  
  --performance-mode generalPurpose \  
  --throughput-mode bursting \  
  --encrypted \  
  --tags Key=Name,Value=my-file-system
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

```
{
  "Key": "Name",
  "Value": "my-file-system"
}
]
```

Per ulteriori informazioni, consulta [Creating Amazon EFS file system](#) nella Amazon Elastic File System User Guide.

- Per API i dettagli, consulta [CreateFileSystem AWS CLI Command Reference](#).

## create-mount-target

Il seguente esempio di codice mostra come utilizzare `create-mount-target`.

### AWS CLI

Per creare un target di montaggio

L'esempio seguente crea una destinazione di montaggio per il file system specificato.

```
aws efs create-mount-target \
  --file-system-id fs-c7a0456e \
  --subnet-id subnet-02bf4c428bexample \
  --security-groups sg-068f739363example
```

Output:

```
{
  "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"
```

```
}
```

Per ulteriori informazioni, consulta [Creazione di destinazioni di montaggio](#) nella Amazon Elastic File System User Guide.

- Per API i dettagli, consulta [CreateMountTarget AWS CLI](#) Command Reference.

## **delete-file-system**

Il seguente esempio di codice mostra come utilizzare `delete-file-system`.

### AWS CLI

Per eliminare un file system

L'`delete-file-system` esempio seguente elimina il file system specificato.

```
aws efs delete-file-system \  
  --file-system-id fs-c7a0456e
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di un EFS file system Amazon](#) nella Amazon Elastic File System User Guide.

- Per API i dettagli, consulta [DeleteFileSystem AWS CLI](#) Command Reference.

## **delete-mount-target**

Il seguente esempio di codice mostra come utilizzare `delete-mount-target`.

### AWS CLI

Per eliminare un target di montaggio

L'`delete-mount-target` esempio seguente elimina il target di montaggio specificato.

```
aws efs delete-mount-target \  
  --mount-target-id fsmt-f9a14450
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creazione di destinazioni di montaggio](#) nella Amazon Elastic File System User Guide.

- Per API i dettagli, consulta [DeleteMountTarget AWS CLI Command Reference](#).

## describe-file-systems

Il seguente esempio di codice mostra come utilizzare `describe-file-systems`.

### AWS CLI

Per descrivere un file system

L'`describe-file-systems` esempio seguente descrive il file system specificato.

```
aws efs describe-file-systems \  
  --file-system-id fs-c7a0456e
```

Output:

```
{  
  "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"
      }
    ]
  }
]
```

Per ulteriori informazioni, consulta [Managing Amazon EFS file system](#) nella Amazon Elastic File System User Guide.

- Per API i dettagli, consulta [DescribeFileSystems AWS CLI Command Reference](#).

## describe-mount-targets

Il seguente esempio di codice mostra come utilizzare `describe-mount-targets`.

### AWS CLI

Per descrivere un target di montaggio

L'`describe-mount-targets` esempio seguente descrive il target di montaggio specificato.

```
aws efs describe-mount-targets \
  --mount-target-id fsmt-f9a14450
```

Output:

```
{
  "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"
      }
    ]
  }
```

Per ulteriori informazioni, consulta [Creazione di destinazioni di montaggio](#) nella Amazon Elastic File System User Guide.

- Per API i dettagli, consulta [DescribeMountTargets AWS CLI Command Reference](#).

## describe-tags

Il seguente esempio di codice mostra come utilizzare `describe-tags`.

### AWS CLI

Per descrivere i tag di un file system

L'`describe-tags`esempio seguente descrive i tag per il file system specificato.

```
aws efs describe-tags \
  --file-system-id fs-c7a0456e
```

Output:

```
{
  "Tags": [
    {
      "Key": "Name",
      "Value": "my-file-system"
    },
    {
      "Key": "Department",
      "Value": "Business Intelligence"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Managing file system tag](#) nella Amazon Elastic File System User Guide.

- Per API i dettagli, consulta [DescribeTags AWS CLI Command Reference](#).

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per recuperare i tag di una risorsa

L'esempio seguente recupera i tag associati al file system specificato.

```
aws efs list-tags-for-resource \  
  --resource-id fs-c7a0456e
```

Output:

```
{  
  "Tags": [  
    {  
      "Key": "Name",  
      "Value": "my-file-system"  
    },  
    {  
      "Key": "Department",  
      "Value": "Business Intelligence"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Managing file system tag](#) nella Amazon Elastic File System User Guide.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare una risorsa

L'`aws efs tag-resource` seguente aggiunge il tag `Department=Business Intelligence` al file system specificato.

```
aws efs tag-resource \  
  --resource-id fs-c7a0456e \  
  --tags Key=Department,Value="Business Intelligence"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing file system tag](#) nella Amazon Elastic File System User Guide.

- Per API i dettagli, consulta [TagResource AWS CLI Command Reference](#).

## aws efs untag-resource

Il seguente esempio di codice mostra come utilizzare `aws efs untag-resource`.

### AWS CLI

Per rimuovere un tag da una risorsa

L'`aws efs untag-resource` seguente rimuove il tag con la chiave del `Department` tag dal file system specificato.

```
aws efs untag-resource \  
  --resource-id fs-c7a0456e \  
  --tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing file system tag](#) nella Amazon Elastic File System User Guide.

- Per API i dettagli, consulta [UntagResource AWS CLI Command Reference](#).

## Esempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con Amazon EKS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **associate-encryption-config**

Il seguente esempio di codice mostra come utilizzare `associate-encryption-config`.

AWS CLI

Per associare una configurazione di crittografia a un cluster esistente

L'esempio seguente abilita la crittografia su EKS cluster esistenti in cui la crittografia non è già abilitata.

```
aws eks associate-encryption-config \  
  --cluster-name my-eks-cluster \  
  --encryption-config '[{"resources":["secrets"],"provider":  
  {"keyArn":"arn:aws:kms:region-code:account:key/key"}}]'
```

Output:

```
{  
  "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": []
}
}

```

Per ulteriori informazioni, consulta la sezione [Abilitazione della crittografia segreta su un cluster esistente](#) nella Amazon EKS User Guide.

- Per API i dettagli, consulta [AssociateEncryptionConfig AWS CLI Command Reference](#).

## associate-identity-provider-config

Il seguente esempio di codice mostra come utilizzare `associate-identity-provider-config`.

### AWS CLI

Associa un provider di identità al tuo Amazon EKS Cluster

L'esempio seguente associa un provider di identità al tuo Amazon EKS Cluster.

```

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

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Autenticare gli utenti del cluster da un provider di identità OpenID Connect - OIDC Associare un provider](#) di identità nella EKS Amazon User Guide.

- Per API i dettagli, consulta AWS CLI Command [AssociateIdentityProviderConfigReference](#).

## create-addon

Il seguente esempio di codice mostra come utilizzare `create-addon`.

### AWS CLI

Esempio 1: creare un EKS componente aggiuntivo Amazon con una versione compatibile predefinita per la rispettiva versione del EKS cluster

Il seguente comando di `create-addon` esempio crea un EKS componente aggiuntivo Amazon con una versione compatibile predefinita per la rispettiva versione del EKS cluster.

```

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

```

Output:

```

{
  "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": {}
  }
}

```

Per ulteriori informazioni, consulta [Managing Amazon EKS Add-ons - Creating an add-on](#) nella Amazon EKS User Guide.

Esempio 2: creare un EKS componente aggiuntivo Amazon con una versione aggiuntiva specifica

Il seguente comando di `create-addon` esempio crea un EKS componente aggiuntivo Amazon con una versione aggiuntiva specifica.

```

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

```

Output:

```

{
  "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": {}
  }
}

```

Per ulteriori informazioni, consulta [Managing Amazon EKS Add-ons - Creating an add-on](#) nella Amazon EKS User Guide.

Esempio 3: creare un EKS componente aggiuntivo Amazon con valori di configurazione personalizzati e risolvere i dettagli dei conflitti

Il seguente comando di `create-addon` esempio crea un EKS componente aggiuntivo Amazon con valori di configurazione personalizzati e risolve i dettagli dei conflitti.

```

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

```

Output:

```

{
  "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\"}}}"
  }
}

```



Per ulteriori informazioni, consulta [Managing Amazon EKS Add-ons - Creating an add-on](#) nella Amazon EKS User Guide.

Esempio 4: creare un EKS componente aggiuntivo Amazon con file di valori JSON di configurazione personalizzati

Il seguente comando di `create-addon` esempio crea un EKS componente aggiuntivo Amazon con valori di configurazione personalizzati e risolve i dettagli dei conflitti.

```
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.json' \  
  --resolve-conflicts OVERWRITE \  
  --tags '{"eks-addon-key-1": "value-1" , "eks-addon-key-2": "value-2"}'
```

Contenuto di `configuration-values.json`.

```
{  
  "resources": {  
    "limits": {  
      "cpu": "150m"  
    }  
  },  
  "env": {  
    "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"  
  }  
}
```

Output:

```
{  
  "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}"
  }
}

```

Per ulteriori informazioni, consulta [Managing Amazon EKS Add-ons - Creating an add-on](#) nella Amazon EKS User Guide.

Esempio 5: creare un EKS componente aggiuntivo Amazon con file di valori YAML di configurazione personalizzati

Il seguente comando di `create-addon` esempio crea un EKS componente aggiuntivo Amazon con valori di configurazione personalizzati e risolve i dettagli dei conflitti.

```

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"}'

```

Contenuto di `configuration-values.yaml`.

```

resources:
  limits:
    cpu: '100m'
env:
  AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'

```

Output:

```
{
  "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:\n
AWS_VPC_K8S_CNI_LOGLEVEL: 'INFO'"
  }
}
```

Per ulteriori informazioni, consulta [Managing Amazon EKS Add-ons - Creating an add-on](#) nella Amazon EKS User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateAddon](#) Reference.

## create-cluster

Il seguente esempio di codice mostra come utilizzare `create-cluster`.

### AWS CLI

Per creare un nuovo cluster

Questo comando di esempio crea un cluster denominato `prod` nella regione predefinita.

Comando:

```
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
```

Output:

```
{
  "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-J7ONKE3BQ4PI",
    "resourcesVpcConfig": {
      "subnetIds": [
        "subnet-6782e71e",
        "subnet-e7e761ac"
      ],
      "securityGroupIds": [
        "sg-6979fe18"
      ],
      "vpcId": "vpc-950809ec"
    },
    "status": "CREATING",
    "certificateAuthority": {}
  }
}
```

Per creare un nuovo cluster con accesso privato agli endpoint e registrazione abilitati

Questo comando di esempio crea un cluster denominato `example` nell'area predefinita con l'accesso pubblico agli endpoint disabilitato, l'accesso privato agli endpoint abilitato e tutti i tipi di registrazione abilitati.

Comando:

```
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-0a188dccd2f9a632f,subnet-09290d93da4278664,subnet-0f21dd86e0e91134a,  
\
```

```
--logging '{"clusterLogging":[{"types":["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'
```

Output:

```
{
  "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"
  }
}

```

- Per API i dettagli, vedere [CreateCluster](#) in AWS CLI Command Reference.

## create-fargate-profile

Il seguente esempio di codice mostra come utilizzare `create-fargate-profile`.

### AWS CLI

Esempio 1: creazione del profilo EKS Fargate per un selettore con un namespace

L'`create-fargate-profile` seguente crea un profilo EKS Fargate per un selettore con un namespace.

```

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"}]'

```

Output:

```

{
  "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": {}
  }
}

```

Per ulteriori informazioni, consulta Profilo [AWS Fargate - Creazione di un profilo Fargate nella Amazon User Guide](#). EKS

Esempio 2: creazione del profilo EKS Fargate per un selettore con namespace ed etichette

L'create-fargate-profileesempio seguente crea un profilo EKS Fargate per un selettore con uno spazio dei nomi e delle etichette.

```

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"}}]'

```

Output:

```

{
  "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": {}
}
}

```

Per ulteriori informazioni, consulta Profilo [AWS Fargate - Creazione di un profilo Fargate nella Amazon User Guide](#). EKS

Esempio 3: Crea il profilo EKS Fargate per un selettore con uno spazio dei nomi e delle etichette, insieme IDs alle sottoreti in cui lanciare un Pod.

L'create-fargate-profileesempio seguente crea il profilo EKS Fargate per un selettore con uno spazio dei nomi e delle etichette, insieme IDs alle sottoreti in cui lanciare un Pod.

```

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"]'

```

Output:

```

{
  "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": {}
}
}

```

Per ulteriori informazioni, consulta Profilo [AWS Fargate - Creazione di un profilo Fargate nella Amazon User Guide](#). EKS

Esempio 4: creazione del profilo EKS Fargate per un selettore con più namespace ed etichette, insieme IDs a sottoreti in cui lanciare un Pod

L'create-fargate-profileesempio seguente crea un profilo EKS Fargate per un selettore con più namespace ed etichette, insieme IDs a sottoreti in cui lanciare un Pod.

```

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"}'

```

Output:

```

{
  "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"
    }
  }
}

```

Per ulteriori informazioni, consulta Profilo [AWS Fargate - Creazione di un profilo Fargate nella Amazon User Guide](#). EKS

Esempio 5: Creazione del profilo EKS Fargate con un selettore di caratteri jolly per namespace ed etichette, insieme IDs alle sottoreti in cui lanciare un Pod

L'`create-fargate-profile` seguente crea un profilo EKS Fargate per un selettore con più namespace ed etichette, insieme IDs a sottoreti in cui lanciare un Pod.

```

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"}'

```

Output:

```

{
  "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"
    }
  }
}

```

Per ulteriori informazioni, consulta Profilo [AWS Fargate - Creazione di un profilo Fargate nella Amazon User Guide](#). EKS

- Per API i dettagli, consulta AWS CLI Command [CreateFargateProfileReference](#).

## create-nodegroup

Il seguente esempio di codice mostra come utilizzare `create-nodegroup`.

### AWS CLI

Esempio 1: crea un gruppo di nodi gestiti per un EKS cluster Amazon

L'esempio seguente crea un gruppo di nodi gestiti per un EKS cluster Amazon.

```
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
```

Output:

```
{
  "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": {}
}
}

```

Per ulteriori informazioni, consulta [Creazione di un gruppo di nodi gestiti](#) nella Amazon EKS User Guide.

Esempio 2: crea un gruppo di nodi gestito per un EKS cluster Amazon con tipi di istanze e dimensioni del disco personalizzati

L'`create-nodesgroup` seguente crea un gruppo di nodi gestiti per un EKS cluster Amazon con tipi di istanze e dimensioni del disco personalizzati.

```

aws eks create-nodesgroup \
  --cluster-name my-eks-cluster \
  --nodesgroup-name my-eks-nodesgroup \
  --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

```

**Output:**

```
{
  "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": {}
  }
}
```

Per ulteriori informazioni, consulta [Creazione di un gruppo di nodi gestiti](#) nella Amazon EKS User Guide.

Esempio 3: crea un gruppo di nodi gestito per un EKS cluster Amazon con tipi di istanza personalizzati, dimensione del disco, tipo ami, tipo di capacità, update-config, etichette, caratteri e tag.

L'create-nodegroupesempio seguente crea un gruppo di nodi gestito per un EKS cluster Amazon con tipi di istanza personalizzati, dimensione del disco, ami-type, capacity-type, update-config, etichette, taints e tag.

```
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"}'
```

Output:

```
{
  "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"
    }
  }
}
```

Per ulteriori informazioni, consulta [Creazione di un gruppo di nodi gestiti](#) nella Amazon EKS User Guide.

- Per API i dettagli, consulta [CreateNodegroup AWS CLI Command Reference](#).



## delete-addon

Il seguente esempio di codice mostra come utilizzare `delete-addon`.

### AWS CLI

Esempio 1. Eliminare un EKS componente aggiuntivo Amazon ma conservare il software aggiuntivo nel Cluster EKS

Il comando di `delete-addon` esempio seguente elimina un EKS componente aggiuntivo Amazon ma conserva il software aggiuntivo nel Cluster. EKS

```
aws eks delete-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon \  
  --preserve
```

Output:

```
{  
  "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": {}  
  }  
}
```

Per ulteriori informazioni, consulta [Gestione dei EKS componenti aggiuntivi di Amazon - Eliminazione di un componente aggiuntivo in Amazon. EKS](#)

Esempio 2. Eliminare un EKS componente aggiuntivo Amazon e anche eliminare il software aggiuntivo dal Cluster EKS

Il comando di `delete-addon` esempio seguente elimina un EKS componente aggiuntivo Amazon ed elimina anche il software aggiuntivo dal Cluster. EKS

```
aws eks delete-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name my-eks-addon
```

Output:

```
{  
  "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": {}  
  }  
}
```

Per ulteriori informazioni, consulta [Gestione dei EKS componenti aggiuntivi di Amazon - Eliminazione di un componente aggiuntivo in Amazon. EKS](#)

- Per API i dettagli, consulta Command [DeleteAddon](#)Reference AWS CLI .

## **delete-cluster**

Il seguente esempio di codice mostra come utilizzare `delete-cluster`.

AWS CLI

Eliminare un piano di controllo del EKS cluster Amazon

L'`delete-cluster` esempio seguente elimina un piano di controllo EKS del cluster Amazon.

```
aws eks delete-cluster \  
  --name my-eks-cluster
```

Output:

```
{  
  "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.y14.us-  
east-2.eks.amazonaws.com",  
    "roleArn": "arn:aws:iam::111122223333:role/eksctl-my-eks-cluster-cluster-  
ServiceRole-zMF6CBakwbW",  
    "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/
DALSJ343KE23J3RN45653DSKJTT647TYD"
  }
},
"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"
}
}
}

```

Per ulteriori informazioni, consulta [Eliminazione di un EKS cluster Amazon](#) nella Amazon EKS User Guide.

- Per API i dettagli, consulta [DeleteCluster AWS CLI Command Reference](#).

## delete-fargate-profile

Il seguente esempio di codice mostra come utilizzare `delete-fargate-profile`.

### AWS CLI

Esempio 1: creazione del profilo EKS Fargate per un selettore con un namespace

L'`delete-fargate-profile` seguente crea un profilo EKS Fargate per un selettore con un namespace.

```
aws eks delete-fargate-profile \  
  --cluster-name my-eks-cluster \  
  --fargate-profile-name my-fargate-profile
```

Output:

```
{  
  "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": {}  
  }  
}
```

Per ulteriori informazioni, consulta [Profilo AWS Fargate - Eliminazione di un Fargate](#) nella Amazon User Guide. EKS

- Per API i dettagli, consulta Command [DeleteFargateProfileReference](#) AWS CLI .

## delete-nodegroup

Il seguente esempio di codice mostra come utilizzare `delete-nodegroup`.

### AWS CLI

Esempio 1: eliminare un gruppo di nodi gestiti per un EKS cluster Amazon

L'`delete-nodegroup` esempio seguente elimina un gruppo di nodi gestiti per un EKS cluster Amazon.

```
aws eks delete-nodegroup \  
  --cluster-name my-eks-cluster \  
  --nodegroup-name my-eks-nodegroup
```

Output:

```
{  
  "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"
    }
}
}
```

- Per API i dettagli, consulta [DeleteNodegroup AWS CLI Command Reference](#).

## deregister-cluster

Il seguente esempio di codice mostra come utilizzare `deregister-cluster`.

## AWS CLI

Annulare la registrazione di un cluster connesso per rimuoverlo dal piano di controllo di Amazon EKS

L'`deregister-cluster` seguente annulla la registrazione di un cluster connesso per rimuoverlo dal piano di controllo di Amazon EKS.

```
aws eks deregister-cluster \  
  --name my-eks-anywhere-cluster
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Annullamento della registrazione di un cluster](#) nella Amazon EKS User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeregisterCluster](#) Reference.

## `describe-addon-configuration`

Il seguente esempio di codice mostra come utilizzare `describe-addon-configuration`.

## AWS CLI

Esempio 1: opzioni di configurazione disponibili durante la creazione o l'aggiornamento di Amazon vpc-cni AddOns



L'output di `aws eks describe-addon-configuration` seguente restituisce tutti gli schemi di configurazione disponibili che usi quando un componente aggiuntivo viene creato o aggiornato per il componente aggiuntivo `vpc-cni` con la rispettiva versione.

```
aws eks describe-addon-configuration \
  --addon-name vpc-cni \
  --addon-version v1.15.1-eksbuild.1
```

Output:

```
{
  "addonName": "vpc-cni",
  "addonVersion": "v1.15.1-eksbuild.1",
  "configurationSchema": "{\n\"$ref\": \"#/definitions/VpcCni\", \"schema\": \"http://\njson-schema.org/draft-06/schema#\", \"definitions\": {\n\"Affinity\": {\n\"type\":\n[\n\"object\", \"null\"]\n}, \"EniConfig\": {\n\"additionalProperties\": false, \"properties\n\": {\n\"create\": {\n\"type\": \"boolean\"}, \"region\": {\n\"type\": \"string\"}, \"subnets\":\n{\n\"additionalProperties\": {\n\"additionalProperties\": false, \"properties\": {\n\"id\":\n{\n\"type\": \"string\"}, \"securityGroups\": {\n\"items\": {\n\"type\": \"string\"}, \"type\":\n\n\"array\"}, \"required\": [\"id\"], \"type\": \"object\", \"minProperties\": 1, \"type\n\": \"object\"}, \"required\": [\"create\", \"region\", \"subnets\"], \"type\": \"object\n\n\", \"Env\": {\n\"additionalProperties\": false, \"properties\": {\n\"ADDITIONAL_ENI_TAGS\n\": {\n\"type\": \"string\"}, \"ANNOTATE_POD_IP\": {\n\"format\": \"boolean\", \"type\":\n\n\"string\"}, \"AWS_EC2_ENDPOINT\": {\n\"type\": \"string\"}, \"AWS_EXTERNAL_SERVICE_CIDRS\n\": {\n\"type\": \"string\"}, \"AWS_MANAGE_ENIS_NON_SCHEDULABLE\": {\n\"format\": \"boolean\n\n\", \"type\": \"string\"}, \"AWS_VPC_CNI_NODE_PORT_SUPPORT\": {\n\"format\": \"boolean\n\n\", \"type\": \"string\"}, \"AWS_VPC_ENI_MTU\": {\n\"format\": \"integer\", \"type\":\n\n\"string\"}, \"AWS_VPC_K8S_CNI_CUSTOM_NETWORK_CFG\": {\n\"format\": \"boolean\", \"type\n\n\": \"string\"}, \"AWS_VPC_K8S_CNI_EXCLUDE_SNAT_CIDRS\": {\n\"type\": \"string\"},\n\n\"AWS_VPC_K8S_CNI_EXTERNALSNAT\": {\n\"format\": \"boolean\", \"type\": \"string\"},\n\n\"AWS_VPC_K8S_CNI_LOGLEVEL\": {\n\"type\": \"string\"}, \"AWS_VPC_K8S_CNI_LOG_FILE\":\n{\n\"type\": \"string\"}, \"AWS_VPC_K8S_CNI_RANDOMIZESNAT\": {\n\"type\": \"string\"},\n\n\"AWS_VPC_K8S_CNI_VETHPREFIX\": {\n\"type\": \"string\"}, \"AWS_VPC_K8S_PLUGIN_LOG_FILE\n\n\": {\n\"type\": \"string\"}, \"AWS_VPC_K8S_PLUGIN_LOG_LEVEL\": {\n\"type\": \"string\n\n\"}, \"CLUSTER_ENDPOINT\": {\n\"type\": \"string\"}, \"DISABLE_INTROSPECTION\":\n{\n\"format\": \"boolean\", \"type\": \"string\"}, \"DISABLE_LEAKED_ENI_CLEANUP\":\n{\n\"format\": \"boolean\", \"type\": \"string\"}, \"DISABLE_METRICS\": {\n\"format\n\n\": \"boolean\", \"type\": \"string\"}, \"DISABLE_NETWORK_RESOURCE_PROVISIONING\n\n\": {\n\"format\": \"boolean\", \"type\": \"string\"}, \"DISABLE_POD_V6\": {\n\"format\n\n\": \"boolean\", \"type\": \"string\"}, \"ENABLE_BANDWIDTH_PLUGIN\": {\n\"format\":\n\n\"boolean\", \"type\": \"string\"}, \"ENABLE_POD_ENI\": {\n\"format\": \"boolean\",\n\n\"type\": \"string\"}, \"ENABLE_PREFIX_DELEGATION\": {\n\"format\": \"boolean\",\n\n\"type\": \"string\"}, \"ENABLE_V4_EGRESS\": {\n\"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\"}"
}

```

Esempio 2: opzioni di configurazione disponibili durante la creazione o l'aggiornamento di Amazon cordns AddOns

L'`describe-addon-configuration` seguente restituisce tutto lo schema di configurazione disponibile che usi quando un componente aggiuntivo viene creato o aggiornato per il componente aggiuntivo `coredns` con la rispettiva versione.

```
aws eks describe-addon-configuration \
  --addon-name coredns \
  --addon-version v1.8.7-eksbuild.4
```

Output:

```
{
  "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\" } } } }
```

Per ulteriori informazioni, consulta [Creazione o aggiornamento di un file kubeconfig per un EKS cluster Amazon](#) in Amazon. EKS

- Per API i dettagli, consulta Command [DescribeAddonConfiguration](#) Reference AWS CLI .

## describe-addon-versions

Il seguente esempio di codice mostra come utilizzare `describe-addon-versions`.

AWS CLI

Esempio 1: Elenca tutti i componenti aggiuntivi disponibili per Cluster EKS

L'output di `describe-addon-versions` seguente elenca tutti i componenti aggiuntivi disponibili.

```
aws eks describe-addon-versions \
  --query 'sort_by(addons &owner)[].{publisher: publisher, owner: owner,
  addonName: addonName, type: type}' \
```

**--output *table***

Output:

```

-----
|                                     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 |
+-----+-----+
+-----+-----+

```

Per ulteriori informazioni, consulta [Managing Amazon EKS Add-ons - Creating an add-on](#) nella Amazon EKS User Guide.

Esempio 2: elenca tutti i componenti aggiuntivi disponibili per la versione di Kubernetes specificata supportata per EKS

L'describe-addon-versionsesempio seguente elenca tutti i componenti aggiuntivi disponibili per la versione di Kubernetes specificata per cui è supportata. 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
```

Output:

```
-----
|                                     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		

```
+-----+-----+
+-----+-----+
```

Per ulteriori informazioni, consulta [Managing Amazon EKS Add-ons - Creating an add-on](#) nella Amazon EKS User Guide.

Esempio 3: elenca tutte le versioni dei componenti aggiuntivi vpc-cni disponibili per la versione di Kubernetes specificata supportata per EKS

L'`describe-addon-versions` seguente elenca tutte le versioni dei componenti aggiuntivi di vpc-cni disponibili per la versione di Kubernetes specificata per cui è supportata. EKS

```
aws eks describe-addon-versions \
  --kubernetes-version=1.26 \
  --addon-name=vpc-cni \
  --query='addons[].addonVersions[].addonVersion'
```

Output:

```
[
  "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"
]
```



Per ulteriori informazioni, consulta [Managing Amazon EKS Add-ons - Creating an add-on](#) nella Amazon EKS User Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeAddonVersionsReference](#).

## describe-addon

Il seguente esempio di codice mostra come utilizzare `describe-addon`.

### AWS CLI

Descrivi l'EKSaddon in esecuzione attiva nel tuo cluster Amazon EKS

L'`describe-addon`esempio seguente esegue attivamente l'EKSaddon nel tuo EKS cluster Amazon.

```
aws eks describe-addon \  
  --cluster-name my-eks-cluster \  
  --addon-name vpc-cni
```

Output:

```
{  
  "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'\nenv:\n  AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'"
```

```
}  
}
```

- Per API i dettagli, consulta [DescribeAddon AWS CLI Command Reference](#).

## describe-cluster

Il seguente esempio di codice mostra come utilizzare `describe-cluster`.

### AWS CLI

Descrivi l'EKSaddon in esecuzione attiva nel tuo cluster Amazon EKS

L'`describe-cluster` esempio seguente esegue attivamente l'EKSaddon nel tuo EKS cluster Amazon.

```
aws eks describe-cluster \  
  --cluster-name my-eks-cluster
```

Output:

```
{  
  "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"
  }
}
}

```

- Per API i dettagli, consulta [DescribeCluster AWS CLI Command Reference](#).

## describe-fargate-profile

Il seguente esempio di codice mostra come utilizzare `describe-fargate-profile`.

### AWS CLI

Descrivi un profilo Fargate

L'`describe-fargate-profile` esempio seguente descrive un profilo Fargate.

```

aws eks describe-fargate-profile \
  --cluster-name my-eks-cluster \
  --fargate-profile-name my-fargate-profile

```

Output:

```

{
  "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"
    }
  }
}

```

- Per API i dettagli, vedere [DescribeFargateProfile](#) in AWS CLI Command Reference.

## describe-identity-provider-config

Il seguente esempio di codice mostra come utilizzare `describe-identity-provider-config`.

### AWS CLI

Descrivi una configurazione di provider di identità associata al tuo Amazon EKS Cluster

L'`describe-identity-provider-config` seguente descrive una configurazione di provider di identità associata al tuo Amazon EKS Cluster.

```
aws eks describe-identity-provider-config \  
  --cluster-name my-eks-cluster \  
  --identity-provider-config type=oidc,name=my-identity-provider
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Autenticare gli utenti del cluster da un provider di identità OpenID Connect](#) nella EKS Amazon User Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeIdentityProviderConfigReference](#).

## describe-nodegroup

Il seguente esempio di codice mostra come utilizzare `describe-nodegroup`.

### AWS CLI

Descrivi un gruppo di nodi gestito per un EKS cluster Amazon

L'esempio seguente descrive un gruppo di nodi gestiti per un EKS cluster Amazon.

```
aws eks describe-nodegroup \
  --cluster-name my-eks-cluster \
  --nodegroup-name my-eks-nodegroup
```

Output:

```
{
  "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": {}
  }
}

```

- Per API i dettagli, consulta [DescribeNodegroup AWS CLI Command Reference](#).

## describe-update

Il seguente esempio di codice mostra come utilizzare `describe-update`.

### AWS CLI

Esempio 1: descrivere un aggiornamento per un cluster

L'`describe-update`esempio seguente descrive un aggiornamento per un cluster denominato.

```

aws eks describe-update \
  --name my-eks-cluster \
  --update-id 10bddb13-a71b-425a-b0a6-71cd03e59161

```

Output:

```

{
  "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": []
  }
}

```

Per ulteriori informazioni, consulta la sezione [Aggiornamento di una versione di Amazon EKS Cluster Kubernetes](#) nella Amazon EKS User Guide.

Esempio 2: Per descrivere un aggiornamento per un cluster

L'`describe-update`esempio seguente descrive un aggiornamento per un cluster denominato.

```

aws eks describe-update \
  --name my-eks-cluster \
  --update-id e4994991-4c0f-475a-a040-427e6da52966

```

Output:

```

{
  "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": []
  }
}

```

Per ulteriori informazioni, consulta la sezione [Aggiornamento di una versione di Amazon EKS Cluster Kubernetes](#) nella Amazon EKS User Guide.

Esempio 3: Per descrivere un aggiornamento per un cluster

L'`describe-update` seguente descrive un aggiornamento per un cluster denominato.

```

aws eks describe-update \
  --name my-eks-cluster \
  --update-id b5f0ba18-9a87-4450-b5a0-825e6e84496f

```

Output:

```

{
  "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": []
  }
}

```

Per ulteriori informazioni, consulta la sezione [Aggiornamento di una versione di Amazon EKS Cluster Kubernetes](#) nella Amazon EKS User Guide.

- Per API i dettagli, consulta Command [DescribeUpdate](#) Reference AWS CLI .

## disassociate-identity-provider-config

Il seguente esempio di codice mostra come utilizzare `disassociate-identity-provider-config`.

### AWS CLI

Dissocia il provider di identità dal tuo Amazon Cluster EKS

L'esempio seguente dissocia un provider di identità dal tuo Amazon EKS Cluster.

```
aws eks disassociate-identity-provider-config \
  --cluster-name my-eks-cluster \
  --identity-provider-config 'type=oidc,name=my-identity-provider'
```

Output:

```
{
  "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": []
  }
}
```

Per ulteriori informazioni, consulta [Autenticare gli utenti del cluster da un provider di identità OpenID Connect - OIDC Dissociare un provider di identità dal cluster](#) nella EKS Amazon User Guide.

- Per API i dettagli, consulta Command [DisassociateIdentityProviderConfig](#) Reference AWS CLI .

## get-token

Il seguente esempio di codice mostra come utilizzare `get-token`.

### AWS CLI

Esempio 1: ottieni un token di autenticazione per un EKS cluster Amazon denominato `my-eks-cluster`

L'`get-token` esempio seguente ottiene un token di autenticazione per un EKS cluster Amazon denominato `my-eks-cluster`.

```
aws eks get-token \  
  --cluster-name my-eks-cluster
```

Output:

```
{  
  "kind": "ExecCredential",  
  "apiVersion": "client.authentication.k8s.io/v1beta1",  
  "spec": {},  
  "status": {  
    "expirationTimestamp": "2024-04-11T20:59:56Z",  
    "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."  
  }  
}
```

Esempio 2: ottiene un token di autenticazione per un EKS cluster Amazon denominato `my-eks-cluster` assumendo questo ruolo ARN per le credenziali al momento della firma del token

L'`get-token` esempio seguente ottiene un token di autenticazione per un Amazon EKS Cluster denominato `my-eks-cluster` assumendo questo ruolo ARN per le credenziali al momento della firma del token.

```
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
```

Output:

```
{
  "kind": "ExecCredential",
  "apiVersion": "client.authentication.k8s.io/v1beta1",
  "spec": {},
  "status": {
    "expirationTimestamp": "2024-04-11T21:05:26Z",
    "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."
  }
}
```

- Per API i dettagli, consulta AWS CLI Command [GetToken](#) Reference.

## list-addons

Il seguente esempio di codice mostra come utilizzare `list-addons`.

### AWS CLI

Elenca tutti i componenti aggiuntivi installati nel tuo EKS cluster Amazon denominati `my-eks-cluster`

L'esempio seguente elenca tutti i componenti aggiuntivi installati nel EKS cluster Amazon denominato `my-eks-cluster`.

```
aws eks list-addons \
  --cluster-name my-eks-cluster
```

Output:

```
{
  "addons": [
    "kube-proxy",
    "vpc-cni"
  ]
}
```

- Per API i dettagli, consulta [ListAddons AWS CLI](#) Command Reference.

## list-clusters

Il seguente esempio di codice mostra come utilizzare `list-clusters`.

## AWS CLI

Per elencare tutti i componenti aggiuntivi installati nel tuo EKS cluster Amazon denominati `` my-eks-cluster

L'`list-clusters` esempio seguente elenca tutti i componenti aggiuntivi installati nel EKS cluster Amazon denominato my-eks-cluster.

```
aws eks list-clusters
```

Output:

```
{
  "clusters": [
    "prod",
    "qa",
    "stage",
    "my-eks-cluster"
  ]
}
```

- Per API i dettagli, consulta [ListClusters AWS CLI Command Reference](#).

## `list-fargate-profiles`

Il seguente esempio di codice mostra come utilizzare `list-fargate-profiles`.

## AWS CLI

Per elencare tutti i profili fargate nel tuo EKS cluster Amazon denominati `` my-eks-cluster

L'`list-fargate-profiles` esempio seguente elenca tutti i profili fargate del EKS cluster Amazon denominato my-eks-cluster.

```
aws eks list-fargate-profiles \
  --cluster-name my-eks-cluster
```

Output:

```
{
```

```
"fargateProfileNames": [  
  "my-fargate-profile"  
]  
}
```

- Per API i dettagli, consulta [ListFargateProfiles AWS CLI Command Reference](#).

## list-identity-provider-configs

Il seguente esempio di codice mostra come utilizzare `list-identity-provider-configs`.

### AWS CLI

Elenca i provider di identità associati a un EKS cluster Amazon

L'`list-identity-provider-configs` seguente elenca i provider di identità associati a un EKS cluster Amazon.

```
aws eks list-identity-provider-configs \  
  --cluster-name my-eks-cluster
```

Output:

```
{  
  "identityProviderConfigs": [  
    {  
      "type": "oidc",  
      "name": "my-identity-provider"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Autenticare gli utenti del cluster da un provider di identità OpenID Connect](#) nella EKS Amazon User Guide.

- Per API i dettagli, consulta AWS CLI Command [ListIdentityProviderConfigs](#) Reference.

## list-nodegroups

Il seguente esempio di codice mostra come utilizzare `list-nodegroups`.

## AWS CLI

Elenca tutti i gruppi di nodi in un EKS cluster Amazon

L'`list-nodegroups`esempio seguente elenca tutti i gruppi di nodi in un EKS cluster Amazon.

```
aws eks list-nodegroups \
  --cluster-name my-eks-cluster
```

Output:

```
{
  "nodegroups": [
    "my-eks-managed-node-group",
    "my-eks-nodegroup"
  ]
}
```

- Per API i dettagli, consulta [ListNodegroups AWS CLI Command Reference](#).

## `list-tags-for-resource`

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

## AWS CLI

Esempio 1: elencare tutti i tag per un EKS cluster Amazon ARN

L'`list-tags-for-resource`esempio seguente elenca tutti i tag per un Amazon EKS ClusterARN.

```
aws eks list-tags-for-resource \
  --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster
```

Output:

```
{
  "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"
  }
}

```

Esempio 2: elencare tutti i tag per un gruppo Amazon EKS Node ARN

L'`list-tags-for-resources` seguente elenca tutti i tag per un gruppo Amazon EKS NodeARN.

```

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

```

Output:

```

{
  "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"
    }
  }
}

```

### Esempio 3: elencare tutti i tag su un profilo Amazon EKS Fargate ARN

L'`list-tags-for-resource` seguente elenca tutti i tag per un profilo Amazon EKS Fargate. ARN

```

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

```

Output:

```

{
  "tags": {
    "eks-fargate-profile-key-2": "value-2",
    "eks-fargate-profile-key-1": "value-1"
  }
}

```

### Esempio 4: elencare tutti i tag per un EKS componente aggiuntivo Amazon ARN

L'`list-tags-for-resource` seguente elenca tutti i tag per un EKS componente aggiuntivo Amazon ARN.

```

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

```

Output:

```

{
  "tags": {
    "eks-addon-key-2": "value-2",
    "eks-addon-key-1": "value-1"
  }
}

```

```
}
```

Esempio 5: elencare tutti i tag per un provider di EKS OIDC identità Amazon ARN

L'`list-tags-for-resource` seguente elenca tutti i tag per un provider di EKS OIDC identità Amazon 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
```

Output:

```
{  
  "tags": {  
    "my-identity-provider": "test"  
  }  
}
```

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## list-update

Il seguente esempio di codice mostra come utilizzare `list-update`.

AWS CLI

Esempio 1: per elencare gli aggiornamenti associati a un nome Amazon EKS Cluster

L'`list-updates` seguente elenca tutti gli aggiornamenti IDs per un nome di Amazon EKS Cluster.

```
aws eks list-updates \  
  --name my-eks-cluster
```

Output:

```
{  
  "updateIds": [  
    "5f78d14e-c57b-4857-a3e4-cf664ae20949",  
  ],  
}
```

```
    "760e5a3f-adad-48c7-88d3-7ac283c09c26",  
    "cd4ec863-bc55-47d5-a377-3971502f529b",  
    "f12657ce-e869-4f17-b158-a82ab8b7d937"  
  ]  
}
```

Esempio 2: elencare tutti gli aggiornamenti IDs per un gruppo Amazon EKS Node

L'`list-updates` esempio seguente elenca tutti gli aggiornamenti IDs per un gruppo Amazon EKS Node.

```
aws eks list-updates \  
  --name my-eks-cluster \  
  --nodegroup-name my-eks-managed-node-group
```

Output:

```
{  
  "updateIds": [  
    "8c6c1bef-61fe-42ac-a242-89412387b8e7"  
  ]  
}
```

Esempio 3: elencare tutti gli aggiornamenti IDs su un EKS componente aggiuntivo Amazon

L'`list-updates` esempio seguente elenca tutti gli aggiornamenti IDs per un EKS componente aggiuntivo Amazon.

```
aws eks list-updates \  
  --name my-eks-cluster \  
  --addon-name vpc-cni
```

Output:

```
{  
  "updateIds": [  
    "9cdba8d4-79fb-3c83-afe8-00b508d33268"  
  ]  
}
```

- Per API i dettagli, consulta [ListUpdate AWS CLI Command Reference](#).

## list-updates

Il seguente esempio di codice mostra come utilizzare `list-updates`.

### AWS CLI

Per elencare gli aggiornamenti per un cluster

Questo comando di esempio elenca gli aggiornamenti correnti per un cluster denominato `example` nell'area predefinita.

Comando:

```
aws eks list-updates --name example
```

Output:

```
{
  "updateIds": [
    "10bddb13-a71b-425a-b0a6-71cd03e59161"
  ]
}
```

- Per API i dettagli, vedere [ListUpdates](#) in AWS CLI Command Reference.

## register-cluster

Il seguente esempio di codice mostra come utilizzare `register-cluster`.

### AWS CLI

Esempio 1: registrare un cluster EKS \_ ANYWHERE Kubernetes esterno su Amazon EKS

L'`register-cluster` esempio seguente registra un ANYWHERE cluster EKS \_ Kubernetes esterno su Amazon. EKS

```
aws eks register-cluster \
  --name my-eks-anywhere-cluster \
  --connector-config 'roleArn=arn:aws:iam::111122223333:role/AmazonEKSClusterAgentRole,provider=EKS_ANYWHERE'
```

Output:

```
{
  "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"
    }
  }
}
```

Per ulteriori informazioni, consulta [Connettere un cluster esterno](#) nella Amazon EKS User Guide.

Esempio 2: registra qualsiasi cluster Kubernetes esterno su Amazon EKS

L'`register-cluster` seguente registra un ANYWHERE cluster EKS \_ Kubernetes esterno su Amazon. EKS

```
aws eks register-cluster \
  --name my-eks-anywhere-cluster \
  --connector-config 'roleArn=arn:aws:iam::111122223333:role/AmazonEKSCollectorAgentRole,provider=OTHER'
```

Output:

```
{
  "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"
    }
}
```

Per ulteriori informazioni, consulta [Connettere un cluster esterno](#) nella Amazon EKS User Guide.

- Per API i dettagli, consulta [RegisterCluster AWS CLI](#) Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Esempio 1: aggiungere i tag specificati a un EKS cluster Amazon

L'`tag-resource` seguente aggiunge i tag specificati a un Amazon EKS Cluster.

```
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'
```

Questo comando non produce alcun output.

Esempio 2: aggiungere i tag specificati a un gruppo Amazon EKS Node

L'`tag-resource` seguente aggiunge i tag specificati a un gruppo Amazon EKS Node.

```
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'
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [TagResource AWS CLI](#) Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

## AWS CLI

Esempio 1: eliminare i tag specificati da un cluster Amazon EKS

L'untag-resource esempio seguente elimina i tag specificati da un Amazon EKS 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"
```

Questo comando non produce alcun output.

Esempio 2: eliminare i tag specificati da un gruppo Amazon EKS Node

L'untag-resource esempio seguente elimina i tag specificati da un gruppo Amazon EKS Node.

```
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"
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [UntagResource AWS CLI Command Reference](#).

## update-addon

Il seguente esempio di codice mostra come utilizzare update-addon.

### AWS CLI

Esempio 1. Per aggiornare un EKS componente aggiuntivo Amazon con il ruolo dell'account di servizio ARN

Il comando di update-addon esempio seguente aggiorna un EKS componente aggiuntivo Amazon con il ruolo ARN dell'account di servizio.

```
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
```



## Output:

```
{
  "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": []
  }
}
```

Per ulteriori informazioni, consulta [Managing Amazon EKS Add-ons - Updating an add-on](#) nella Amazon EKS User Guide.

Esempio 2. Per aggiornare un EKS componente aggiuntivo Amazon con una versione aggiuntiva specifica

Il comando di update-addon esempio seguente aggiorna un EKS componente aggiuntivo Amazon con una versione aggiuntiva specifica.

```
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
```

## Output:

```
{
  "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": []
  }
}

```

Per ulteriori informazioni, consulta [Managing Amazon EKS Add-ons - Updating an add-on](#) nella Amazon EKS User Guide.

Esempio 3. Per aggiornare un EKS componente aggiuntivo Amazon con valori di configurazione personalizzati e risolvere i dettagli dei conflitti

Il seguente comando di `update-addon` esempio aggiorna un EKS componente aggiuntivo Amazon con valori di configurazione personalizzati e risolve i dettagli dei conflitti.

```

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

```

Output:

```

{
  "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": []
  }
}

```

Per ulteriori informazioni, consulta [Managing Amazon EKS Add-ons - Updating an add-on](#) nella Amazon EKS User Guide.

Esempio 4. Per aggiornare un EKS componente aggiuntivo Amazon con un file di valori JSON di configurazione personalizzati

Il seguente comando di `update-addon` esempio aggiorna un EKS componente aggiuntivo Amazon con valori di JSON configurazione personalizzati e risolve i dettagli dei conflitti.

```

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

```

## Contenuto di configuration-values.json.

```
{
  "resources": {
    "limits": {
      "cpu": "100m"
    },
    "requests": {
      "cpu": "50m"
    }
  },
  "env": {
    "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"
  }
}
```

## Output:

```
{
  "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
      \"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": []
}
}

```

Per ulteriori informazioni, consulta [Managing Amazon EKS Add-ons - Updating an add-on](#) nella Amazon EKS User Guide.

Esempio 5. Per aggiornare un EKS componente aggiuntivo Amazon con un file di valori YAML di configurazione personalizzati

Il seguente comando di `update-addon` esempio aggiorna un EKS componente aggiuntivo Amazon con valori di YAML configurazione personalizzati e risolve i dettagli dei conflitti.

```

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

```

Contenuto di `configuration-values.yaml`.

```

resources:
  limits:
    cpu: '100m'
  requests:
    cpu: '50m'
env:
  AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'

```

Output:

```

{
  "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": []
  }
}

```

Per ulteriori informazioni, consulta [Managing Amazon EKS Add-ons - Updating an add-on](#) nella Amazon EKS User Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateAddon](#)Reference.

## update-cluster-config

Il seguente esempio di codice mostra come utilizzare `update-cluster-config`.

### AWS CLI

Per aggiornare l'accesso agli endpoint del cluster

Questo comando di esempio aggiorna un cluster per disabilitare l'accesso pubblico agli endpoint e abilitare l'accesso privato agli endpoint.

Comando:

```
aws eks update-cluster-config --name example \
--resources-vpc-config endpointPublicAccess=false,endpointPrivateAccess=true
```

Output:

```
{
  "update": {
    "id": "ec883c93-2e9e-407c-a22f-8f6fa6e67d4f",
    "status": "InProgress",
    "type": "EndpointAccessUpdate",
    "params": [
      {
        "type": "EndpointPublicAccess",
        "value": "false"
      },
      {
        "type": "EndpointPrivateAccess",
        "value": "true"
      }
    ],
    "createdAt": 1565806986.506,
    "errors": []
  }
}
```

Per abilitare la registrazione per un cluster

Questo comando di esempio abilita tutti i tipi di registrazione del piano di controllo del cluster per un cluster denominato. *example*

Comando:

```
aws eks update-cluster-config --name example \
--logging '{"clusterLogging":[{"types":
["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'
```

Output:

```
{
  "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": []
  }
}

```

- Per API i dettagli, vedere [UpdateClusterConfig](#) in AWS CLI Command Reference.

## update-cluster-version

Il seguente esempio di codice mostra come utilizzare `update-cluster-version`.

### AWS CLI

Per aggiornare un EKS cluster Amazon denominato `my-eks-cluster` alla versione Kubernetes specificata

L'`update-cluster-version`esempio seguente aggiorna un EKS cluster Amazon alla versione Kubernetes specificata.

```

aws eks update-cluster-version \
  --name my-eks-cluster \
  --kubernetes-version 1.27

```

Output:

```

{
  "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": []
  }
}
```

Per ulteriori informazioni, consulta la sezione [Aggiornamento di una versione di Amazon EKS Cluster Kubernetes](#) nella Amazon EKS User Guide.

- Per API i dettagli, consulta Command [UpdateClusterVersion](#) Reference AWS CLI .

## update-kubeconfig

Il seguente esempio di codice mostra come utilizzare `update-kubeconfig`.

### AWS CLI

Esempio 1: configura il tuo kubectl creando o aggiornando kubeconfig in modo da poterti connettere a un cluster Amazon denominato `my-eks-cluster`

L'update-kubeconfig esempio seguente configura il tuo kubectl creando o aggiornando kubeconfig in modo che tu possa connetterti a un Amazon Cluster denominato. `my-eks-cluster`

```
aws eks update-kubeconfig \
  --name my-eks-cluster
```

Output:

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/
xxx/.kube/config
```

Per ulteriori informazioni, consulta [Creazione o aggiornamento di un file kubeconfig per un EKS cluster Amazon nella Amazon EKS User Guide](#).

Esempio 2: configura il tuo kubectl creando o aggiornando kubeconfig (con l'opzione `role-arn` per assumere un ruolo per l'autenticazione del cluster) in modo da poterti connettere a un cluster Amazon chiamato `my-eks-cluster`

L'update-kubeconfigesempio seguente configura il tuo kubectl creando o aggiornando kubeconfig (con l'opzione `role-arn` per assumere un ruolo per l'autenticazione del cluster) in modo che tu possa connetterti a un Amazon Cluster denominato. 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
```

Output:

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/  
xxx/.kube/config
```

Per ulteriori informazioni, consulta [Creazione o aggiornamento di un file kubeconfig per un EKS cluster Amazon nella Amazon EKS User Guide](#).

Esempio 3: configura il tuo kubectl creando o aggiornando kubeconfig (con l'opzione `role-arn` per assumere un ruolo per l'autenticazione del cluster insieme a `alias cluster` e `alias utente personalizzati`) in modo da poterti connettere a un cluster Amazon chiamato `my-eks-cluster`

L'update-kubeconfigesempio seguente configura il tuo kubectl creando o aggiornando kubeconfig (con l'opzione `role-arn` per assumere un ruolo per l'autenticazione del cluster insieme a `alias cluster` e `alias utente personalizzati`) in modo da poterti connettere a un Amazon Cluster denominato. 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
```

Output:

```
Updated context stage-eks-cluster in /Users/dubaria/.kube/config
```

Per ulteriori informazioni, consulta [Creazione o aggiornamento di un file kubeconfig per un EKS cluster Amazon nella Amazon EKS User Guide](#).

Esempio 4: stampa le voci del file kubeconfig per la revisione e configura il tuo kubectl in modo da poterti connettere a un cluster Amazon denominato `` EKS my-eks-cluster

L'update-kubeconfigesempio seguente configura il tuo kubectl creando o aggiornando kubeconfig (con l'opzione role-arn per assumere un ruolo per l'autenticazione del cluster insieme a alias cluster e alias utente personalizzati) in modo da poterti connettere a un Amazon Cluster denominato. 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
```

Output:

```
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

```

Per ulteriori informazioni, consulta [Creazione o aggiornamento di un file kubeconfig per un EKS cluster Amazon nella Amazon EKS User Guide](#).

- Per API i dettagli, consulta Command [UpdateKubeconfig](#) Reference AWS CLI .

## update-nodegroup-config

Il seguente esempio di codice mostra come utilizzare `update-nodegroup-config`.

### AWS CLI

Esempio 1: aggiorna un gruppo di nodi gestiti per aggiungere nuove etichette e colorare il nodo di EKS lavoro per un cluster Amazon EKS

L'`update-nodegroup-config`esempio seguente aggiorna un gruppo di nodi gestiti per aggiungere nuove etichette e contaminazioni al nodo di EKS lavoro per un EKS cluster Amazon.

```

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}]'

```

Output:

```

{
  "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": []
  }
}

```

Per ulteriori informazioni, consulta la sezione [Aggiornamento di un gruppo di nodi gestiti](#) nella Amazon EKS User Guide.

Esempio 2: aggiorna un gruppo di nodi gestito per rimuovere etichette e contaminazioni per il nodo di EKS lavoro per un cluster Amazon EKS

L'update-nodegroup-configesempio seguente aggiorna un gruppo di nodi gestiti per rimuovere etichette e contaminazioni per il nodo di EKS lavoro per un EKS cluster Amazon.

```

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}]'

```

Output:

```

{
  "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": []
}
}

```

Per ulteriori informazioni, consulta la sezione [Aggiornamento di un gruppo di nodi gestiti](#) nella Amazon EKS User Guide.

Esempio 3: aggiorna un gruppo di nodi gestiti per rimuovere e aggiungere etichette e contaminazioni per il nodo di EKS lavoro per un cluster Amazon EKS

L'update-nodegroup-configesempio seguente aggiorna un gruppo di nodi gestiti per rimuovere e aggiungere etichette e contaminazioni per il nodo di EKS lavoro per un EKS cluster Amazon.

```

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}]'

```

Output:

```

{
  "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": []
  }
}

```

Per ulteriori informazioni, consulta la sezione [Aggiornamento di un gruppo di nodi gestiti](#) nella Amazon EKS User Guide.

Esempio 4: aggiorna un gruppo di nodi gestito per aggiornare scaling-config e update-config per il nodo di lavoro EKS per un cluster Amazon EKS

L'update-nodegroup-configesempio seguente aggiorna un gruppo di nodi gestito per aggiornare scaling-config e update-config per il EKS nodo di lavoro per un cluster Amazon. 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

```

**Output:**

```
{
  "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": []
  }
}
```

Per ulteriori informazioni, consulta la sezione [Aggiornamento di un gruppo di nodi gestiti](#) nella Amazon EKS User Guide.

- Per API i dettagli, consulta [UpdateNodegroupConfig AWS CLI Command Reference](#).

**update-nodegroup-version**

Il seguente esempio di codice mostra come utilizzare `update-nodegroup-version`.

**AWS CLI**

Esempio 1: aggiorna la versione di Kubernetes o la versione di un AMI gruppo di nodi gestiti da Amazon EKS



L'update-nodegroup-versionesempio seguente aggiorna la versione Kubernetes o AMI la versione di un gruppo di nodi EKS gestiti di Amazon all'ultima versione disponibile per il tuo cluster Kubernetes.

```
aws eks update-nodegroup-version \  
  --cluster-name my-eks-cluster \  
  --nodegroup-name my-eks-nodegroup \  
  --no-force
```

Output:

```
{  
  "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": []  
  }  
}
```

Per ulteriori informazioni, consulta la sezione [Aggiornamento di un gruppo di nodi gestiti](#) nella Amazon EKS User Guide.

Esempio 2: aggiorna la versione di Kubernetes o la versione di un AMI gruppo di nodi gestiti da Amazon EKS

L'update-nodegroup-versionesempio seguente aggiorna la versione di Kubernetes o la versione di un gruppo di nodi EKS gestiti da Amazon alla AMI versione di rilascio specificataAMI.

```
aws eks update-nodegroup-version \  
  --cluster-name my-eks-cluster \  
  --nodegroup-name my-eks-nodegroup \  
  --no-force
```

```
--nodegroup-name my-eks-nodegroup \  
--kubernetes-version '1.26' \  
--release-version '1.26.12-20240307' \  
--no-force
```

Output:

```
{  
  "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": []  
  }  
}
```

Per ulteriori informazioni, consulta la sezione Aggiornamento di un gruppo di nodi gestiti - <<https://docs.aws.amazon.com/eks/latest/userguide/update-managed-node-group.html>>` nella Amazon User Guide. EKS

- Per i API dettagli, consulta [UpdateNodegroupVersion](#) Command Reference.AWS CLI

## Esempi di utilizzo di Elastic Beanstalk AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface con Elastic Beanstalk.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **abort-environment-update**

Il seguente esempio di codice mostra come utilizzare `abort-environment-update`.

#### AWS CLI

Per interrompere una distribuzione

Il comando seguente interrompe la distribuzione di una versione dell'applicazione in esecuzione per un ambiente denominato: `my-env`

```
aws elasticbeanstalk abort-environment-update --environment-name my-env
```

- Per API i dettagli, vedere [AbortEnvironmentUpdate](#) in AWS CLI Command Reference.

### **check-dns-availability**

Il seguente esempio di codice mostra come utilizzare `check-dns-availability`.

#### AWS CLI

Per verificare la disponibilità di un CNAME

Il comando seguente verifica la disponibilità del sottodominio: `my-cname.elasticbeanstalk.com`

```
aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
```

Output:

```
{
```

```
"Available": true,  
"FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"  
}
```

- Per API i dettagli, vedere [CheckDnsAvailability](#) in AWS CLI Command Reference.

## create-application-version

Il seguente esempio di codice mostra come utilizzare `create-application-version`.

### AWS CLI

Per creare una nuova versione dell'applicazione

Il comando seguente crea una nuova versione, «v1», di un'applicazione denominata «: 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
```

L'applicazione verrà creata automaticamente se non esiste già, grazie all' `auto-create-application` opzione. Il pacchetto sorgente è un file.war memorizzato in un bucket s3 denominato «my-bucket» che contiene l'applicazione di esempio Apache Tomcat.

Output:

```
{  
  "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 Per i AWS CLI dettagli, consulta Command Reference. [CreateApplicationVersion](#)

## create-application

Il seguente esempio di codice mostra come utilizzare `create-application`.

### AWS CLI

Per creare una nuova applicazione

Il comando seguente crea una nuova applicazione denominata "MyApp«:

```
aws elasticbeanstalk create-application --application-name MyApp --description "my application"
```

Il `create-application` comando configura solo il nome e la descrizione dell'applicazione. Per caricare il codice sorgente dell'applicazione, create una versione iniziale dell'applicazione utilizzando `create-application-version`. `create-application-version` dispone inoltre di un'auto-`create-application` opzione che consente di creare l'applicazione e la versione dell'applicazione in un unico passaggio.

Output:

```
{
  "Application": {
    "ApplicationName": "MyApp",
    "ConfigurationTemplates": [],
    "DateUpdated": "2015-02-12T18:32:21.181Z",
    "Description": "my application",
    "DateCreated": "2015-02-12T18:32:21.181Z"
  }
}
```

- Per API i dettagli, vedere [CreateApplication](#) in AWS CLI Command Reference.

## create-configuration-template

Il seguente esempio di codice mostra come utilizzare `create-configuration-template`.

### AWS CLI

Per creare un modello di configurazione

Il comando seguente crea un modello di configurazione denominato in base `my-app-v1` alle impostazioni applicate a un ambiente con l'`ide-rpqsewtp2j`:

```
aws elasticbeanstalk create-configuration-template --application-name my-app --  
template-name my-app-v1 --environment-id e-rpqsewtp2j
```

Output:

```
{  
  "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"  
}
```

- Per API i dettagli, vedere [CreateConfigurationTemplate](#) in AWS CLI Command Reference.

## create-environment

Il seguente esempio di codice mostra come utilizzare `create-environment`.

### AWS CLI

Per creare un nuovo ambiente per un'applicazione

Il comando seguente crea un nuovo ambiente per la versione «v1» di un'applicazione java denominata «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"
```

Output:

```
{  
  "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 è l'etichetta di una versione dell'applicazione precedentemente caricata con `create-application-version`

Specificare un JSON file per definire le opzioni di configurazione dell'ambiente

Il `create-environment` comando seguente specifica che `myoptions.json` deve essere utilizzato un JSON file con lo stesso nome per sovrascrivere i valori ottenuti dallo stack di soluzioni o dal modello di configurazione:

```

aws elasticbeanstalk create-environment --environment-name sample-env --application-
name sampleapp --option-settings file://myoptions.json

```

`myoptions.json` è un JSON oggetto che definisce diverse impostazioni:

```

[
  {
    "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"
  }
]
```

Per ulteriori informazioni, consulta [Option Values](#) nella [AWS Elastic Beanstalk Developer Guide](#).

- Per API i dettagli, consulta [Command `CreateEnvironment` Reference AWS CLI](#).

## **create-storage-location**

Il seguente esempio di codice mostra come utilizzare `create-storage-location`.

### AWS CLI

Per creare una posizione di archiviazione

Il comando seguente crea una posizione di archiviazione in Amazon S3:

```
aws elasticbeanstalk create-storage-location
```

Output:

```
{
  "S3Bucket": "elasticbeanstalk-us-west-2-0123456789012"
}
```

- Per API i dettagli, consulta [CreateStorageLocation AWS CLI Command Reference](#).

## **delete-application-version**

Il seguente esempio di codice mostra come utilizzare `delete-application-version`.

### AWS CLI

Per eliminare la versione di un'applicazione

Il comando seguente elimina una versione dell'applicazione denominata `22a0-stage-150819_182129` per un'applicazione denominata `my-app`:

```
aws elasticbeanstalk delete-application-version --version-label 22a0-stage-150819_182129 --application-name my-app
```



- Per API i dettagli, vedere [DeleteApplicationVersion](#) in AWS CLI Command Reference.

## delete-application

Il seguente esempio di codice mostra come utilizzare `delete-application`.

### AWS CLI

Eliminazione di un'applicazione

Il comando seguente elimina un'applicazione denominata `my-app`:

```
aws elasticbeanstalk delete-application --application-name my-app
```

- Per API i dettagli, vedere [DeleteApplication](#) in AWS CLI Command Reference.

## delete-configuration-template

Il seguente esempio di codice mostra come utilizzare `delete-configuration-template`.

### AWS CLI

Per eliminare un modello di configurazione

Il comando seguente elimina un modello di configurazione denominato `my-template` per un'applicazione denominata `my-app`:

```
aws elasticbeanstalk delete-configuration-template --template-name my-template --application-name my-app
```

- Per API i dettagli, vedere [DeleteConfigurationTemplate](#) in AWS CLI Command Reference.

## delete-environment-configuration

Il seguente esempio di codice mostra come utilizzare `delete-environment-configuration`.

### AWS CLI

Per eliminare una bozza di configurazione

Il comando seguente elimina una bozza di configurazione per un ambiente denominato `my-env`:

```
aws elasticbeanstalk delete-environment-configuration --environment-name my-env --  
application-name my-app
```

- Per API i dettagli, vedere [DeleteEnvironmentConfiguration](#) in AWS CLI Command Reference.

## describe-application-versions

Il seguente esempio di codice mostra come utilizzare `describe-application-versions`.

### AWS CLI

Per visualizzare informazioni sulla versione di un'applicazione

Il comando seguente recupera le informazioni sulla versione di un'applicazione v2 etichettata:

```
aws elasticbeanstalk describe-application-versions --application-name my-app --  
version-label "v2"
```

Output:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

```
    }  
  }  
]  
}
```

- Per API i dettagli, vedere [DescribeApplicationVersions](#) in AWS CLI Command Reference.

## describe-applications

Il seguente esempio di codice mostra come utilizzare `describe-applications`.

### AWS CLI

Per visualizzare un elenco di applicazioni

Il comando seguente recupera informazioni sulle applicazioni nell'area corrente:

```
aws elasticbeanstalk describe-applications
```

Output:

```
{  
  "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"
    }
  ]
}

```

- Per API i dettagli, vedere [DescribeApplications](#) in AWS CLI Command Reference.

## describe-configuration-options

Il seguente esempio di codice mostra come utilizzare `describe-configuration-options`.

### AWS CLI

Per visualizzare le opzioni di configurazione per un ambiente

Il comando seguente recupera le descrizioni di tutte le opzioni di configurazione disponibili per un ambiente denominato `my-env`:

```
aws elasticbeanstalk describe-configuration-options --environment-name my-env --
application-name my-app
```

Output (abbreviato):

```

{
  "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"
    }
]
}

```

Le opzioni di configurazione disponibili variano in base alla piattaforma e alla versione di configurazione. Per ulteriori informazioni sugli spazi dei nomi e sulle opzioni supportate, consulta [Option Values](#) nella Elastic AWS Beanstalk Developer Guide.

- Per i API dettagli, consulta Command Reference. [DescribeConfigurationOptions](#) AWS CLI

## describe-configuration-settings

Il seguente esempio di codice mostra come utilizzare `describe-configuration-settings`.

### AWS CLI

Per visualizzare le impostazioni di configurazione per un ambiente

Il comando seguente recupera le impostazioni di configurazione per un ambiente denominato: `my-env`

```
aws elasticbeanstalk describe-configuration-settings --environment-name my-env --  
application-name my-app
```

Output (abbreviato):

```
{  
  "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"
    }
  ]
}

```

Per ulteriori informazioni sugli spazi dei nomi e sulle opzioni supportate, consulta Option Values nella Elastic AWS Beanstalk Developer Guide.

- Per i API dettagli, consulta Command Reference. [DescribeConfigurationSettings](#) AWS CLI

## describe-environment-health

Il seguente esempio di codice mostra come utilizzare `describe-environment-health`.

### AWS CLI

Per visualizzare lo stato dell'ambiente

Il comando seguente recupera le informazioni generali sullo stato di salute per un ambiente denominato `my-env`:

```
aws elasticbeanstalk describe-environment-health --environment-name my-env --
attribute-names ALL
```

Output:

```

{
  "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": []
}
```

Le informazioni sanitarie sono disponibili solo per gli ambienti in cui è abilitata la reportistica sanitaria avanzata. Per ulteriori informazioni, consulta [Enhanced Health Reporting and Monitoring](#) nella [AWS Elastic Beanstalk Developer Guide](#).

- Per API i dettagli, consulta Command [DescribeEnvironmentHealth](#) Reference AWS CLI .

## describe-environment-resources

Il seguente esempio di codice mostra come utilizzare `describe-environment-resources`.

### AWS CLI

Per visualizzare informazioni sulle AWS risorse presenti nell'ambiente

Il comando seguente recupera informazioni sulle risorse in un ambiente denominato `my-env`:

```
aws elasticbeanstalk describe-environment-resources --environment-name my-env
```



**Output:**

```
{
  "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"
      }
    ]
  }
}
```

- Per API i dettagli, vedere [DescribeEnvironmentResources](#) in AWS CLI Command Reference.

**describe-environments**

Il seguente esempio di codice mostra come utilizzare `describe-environments`.

**AWS CLI**

Per visualizzare informazioni su un ambiente

Il comando seguente recupera informazioni su un ambiente denominato `my-env`:

```
aws elasticbeanstalk describe-environments --environment-names my-env
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeEnvironments](#) in AWS CLI Command Reference.

## describe-events

Il seguente esempio di codice mostra come utilizzare `describe-events`.

AWS CLI

Per visualizzare gli eventi relativi a un ambiente

Il comando seguente recupera gli eventi per un ambiente denominato `my-env`:

```
aws elasticbeanstalk describe-events --environment-name my-env
```

Output (abbreviato):

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeEvents](#) in AWS CLI Command Reference.

## describe-instances-health

Il seguente esempio di codice mostra come utilizzare `describe-instances-health`.

### AWS CLI

Per visualizzare lo stato dell'ambiente

Il comando seguente recupera le informazioni sullo stato delle istanze in un ambiente denominato: `my-env`

```
aws elasticbeanstalk describe-instances-health --environment-name my-env --  
attribute-names ALL
```

Output:

```
{  
  "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,  
          0.0,  
          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"  
}
```

Le informazioni sanitarie sono disponibili solo per gli ambienti in cui è abilitata la reportistica sanitaria avanzata. Per ulteriori informazioni, consulta [Enhanced Health Reporting and Monitoring](#) nella [AWS Elastic Beanstalk Developer Guide](#).

- Per API i dettagli, consulta [Command `DescribeInstancesHealth`](#) Reference AWS CLI .

## **list-available-solution-stacks**

Il seguente esempio di codice mostra come utilizzare `list-available-solution-stacks`.

### AWS CLI

Per visualizzare gli stack di soluzioni

Il comando seguente elenca gli stack di soluzioni per tutte le configurazioni di piattaforma attualmente disponibili e per quelle utilizzate in passato:

```
aws elasticbeanstalk list-available-solution-stacks
```

Output (abbreviato):

```

{
  "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"
    },
    ...
  ]
}

```

```
}
```

- Per API i dettagli, vedere [ListAvailableSolutionStacks](#) in AWS CLI Command Reference.

## **rebuild-environment**

Il seguente esempio di codice mostra come utilizzare `rebuild-environment`.

### AWS CLI

Per ricostruire un ambiente

Il comando seguente termina e ricrea le risorse in un ambiente denominato: `my-env`

```
aws elasticbeanstalk rebuild-environment --environment-name my-env
```

- Per API i dettagli, vedere [RebuildEnvironment](#) in AWS CLI Command Reference.

## **request-environment-info**

Il seguente esempio di codice mostra come utilizzare `request-environment-info`.

### AWS CLI

Per richiedere registri in coda

Il comando seguente richiede i log da un ambiente denominato: `my-env`

```
aws elasticbeanstalk request-environment-info --environment-name my-env --info-type tail
```

Dopo aver richiesto i log, recupera la loro posizione con `retrieve-environment-info`

- Per API i dettagli, consultate Command [RequestEnvironmentInfo](#) Reference AWS CLI .

## **restart-app-server**

Il seguente esempio di codice mostra come utilizzare `restart-app-server`.

### AWS CLI

Per riavviare i server delle applicazioni

Il comando seguente riavvia i server delle applicazioni su tutte le istanze in un ambiente denominato: `my-env`

```
aws elasticbeanstalk restart-app-server --environment-name my-env
```

- Per API i dettagli, vedere [RestartAppServer](#) in AWS CLI Command Reference.

## retrieve-environment-info

Il seguente esempio di codice mostra come utilizzare `retrieve-environment-info`.

### AWS CLI

Per recuperare tronchi in coda

Il comando seguente recupera un collegamento ai log da un ambiente denominato: `my-env`

```
aws elasticbeanstalk retrieve-environment-info --environment-name my-env --info-type tail
```

Output:

```
{
  "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%2BEa10V6A2HI0x4Rcfb7LT16bBM%3D",
      "InfoType": "tail",
      "Ec2InstanceId": "i-09c1c867"
    }
  ]
}
```

Visualizzate il collegamento in un browser. Prima del recupero, i log devono essere richiesti con `request-environment-info`

- Per API i dettagli, vedere [RetrieveEnvironmentInfo](#) in AWS CLI Command Reference.



## swap-environment-cnames

Il seguente esempio di codice mostra come utilizzare `swap-environment-cnames`.

### AWS CLI

Per scambiare l'ambiente CNAMEs

Il comando seguente scambia i sottodomini assegnati di due ambienti:

```
aws elasticbeanstalk swap-environment-cnames --source-environment-name my-env-blue
--destination-environment-name my-env-green
```

- Per API i dettagli, vedere [SwapEnvironmentCnames](#) in AWS CLI Command Reference.

## terminate-environment

Il seguente esempio di codice mostra come utilizzare `terminate-environment`.

### AWS CLI

Per terminare un ambiente

Il comando seguente termina un ambiente Elastic Beanstalk denominato: `my-env`

```
aws elasticbeanstalk terminate-environment --environment-name my-env
```

Output:

```
{
  "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"
}

```

- Per i API dettagli, consulta [TerminateEnvironment](#) Command Reference.AWS CLI

## update-application-version

Il seguente esempio di codice mostra come utilizzare `update-application-version`.

### AWS CLI

Per modificare la descrizione della versione di un'applicazione

Il comando seguente aggiorna la descrizione di una versione dell'applicazione denominata `22a0-stage-150819_185942`:

```

aws elasticbeanstalk update-application-version --version-label 22a0-stage-150819_185942 --application-name my-app --description "new description"

```

Output:

```

{
  "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"
    }
  }
}

```

- Per API i dettagli, vedere [UpdateApplicationVersion](#) in AWS CLI Command Reference.

## update-application

Il seguente esempio di codice mostra come utilizzare `update-application`.

### AWS CLI

Per modificare la descrizione di un'applicazione

Il comando seguente aggiorna la descrizione di un'applicazione denominata `my-app`:

```
aws elasticbeanstalk update-application --application-name my-app --description "my Elastic Beanstalk application"
```

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [UpdateApplication](#) in AWS CLI Command Reference.

## update-configuration-template

Il seguente esempio di codice mostra come utilizzare `update-configuration-template`.

### AWS CLI

Per aggiornare un modello di configurazione

Il comando seguente rimuove la configurazione delle metriche sanitarie CloudWatch personalizzate configurate ConfigDocument da un modello di configurazione salvato denominato `my-template`:

```
aws elasticbeanstalk update-configuration-template --template-name my-template --application-name my-app --options-to-remove Namespace=aws:elasticbeanstalk:healthreporting:system,OptionName=ConfigDocument
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni sugli spazi dei nomi e sulle opzioni supportate, consulta Option Values nella Elastic AWS Beanstalk Developer Guide.

- Per i API dettagli, consulta Command Reference. [UpdateConfigurationTemplate](#) AWS CLI

## update-environment

Il seguente esempio di codice mostra come utilizzare `update-environment`.

### AWS CLI

Per aggiornare un ambiente a una nuova versione

Il comando seguente aggiorna un ambiente chiamato «my-env» alla versione «v2» dell'applicazione a cui appartiene:

```
aws elasticbeanstalk update-environment --environment-name my-env --version-label v2
```

Questo comando richiede che l'ambiente «my-env» esista già e appartenga a un'applicazione con una versione valida dell'applicazione con l'etichetta «v2».

Output:

```
{
  "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"
}
```

Per impostare una variabile di ambiente

Il comando seguente imposta il valore della variabile "PARAM1" nell'ambiente «my-env» su "«: ParamValue

```
aws elasticbeanstalk update-environment --environment-name my-env --option-
settings Namespace=aws:elasticbeanstalk:application:environment,OptionName=PARAM1,Value=Para
```

Il option-settings parametro richiede uno spazio dei nomi oltre al nome e al valore della variabile. Elastic Beanstalk supporta diversi namespace per le opzioni oltre alle variabili di ambiente.

Per configurare le impostazioni delle opzioni da un file

Il comando seguente configura diverse opzioni nello spazio dei `aws:elb:loadbalancer` nomi da un file:

```
aws elasticbeanstalk update-environment --environment-name my-env --option-
settings file://options.json
```

`options.json` è un JSON oggetto che definisce diverse impostazioni:

```
[
  {
    "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"
  }
]
```

### Output:

```
{
  "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"
  }

```

Per ulteriori informazioni sugli spazi dei nomi e sulle opzioni supportate, consulta [Option Values](#) nella [Elastic AWS Beanstalk Developer Guide](#).

- Per i API dettagli, consulta [Command Reference](#). [UpdateEnvironment](#) AWS CLI

## validate-configuration-settings

Il seguente esempio di codice mostra come utilizzare `validate-configuration-settings`.

### AWS CLI

Per convalidare le impostazioni di configurazione

Il comando seguente convalida un documento di configurazione delle metriche CloudWatch personalizzato:

```

aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings file://options.json

```

`options.json` è un JSON documento che include una o più impostazioni di configurazione da convalidare:

```

[
  {
    "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}"
  }
]
```

Se le opzioni specificate sono valide per l'ambiente specificato, Elastic Beanstalk restituisce un array Messages vuoto:

```
{
  "Messages": []
}
```

Se la convalida fallisce, la risposta includerà informazioni sull'errore:

```
{
  "Messages": [
    {
      "OptionName": "ConfigDocumet",
      "Message": "Invalid option specification (Namespace:
'aws:elasticbeanstalk:healthreporting:system', OptionName: 'ConfigDocumet'):
Unknown configuration setting.",
      "Namespace": "aws:elasticbeanstalk:healthreporting:system",
      "Severity": "error"
    }
  ]
}
```

Per ulteriori informazioni sugli spazi dei nomi e sulle opzioni supportate, consulta Option Values nella Elastic AWS Beanstalk Developer Guide.

- Per i API dettagli, consulta Command Reference. [ValidateConfigurationSettings](#) AWS CLI

## Elastic Load Balancing - Esempi di utilizzo della versione 1 AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Elastic Load Balancing - Versione 1.



Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **add-tags**

Il seguente esempio di codice mostra come utilizzare `add-tags`.

AWS CLI

Per aggiungere un tag a un sistema di bilanciamento del carico

Questo esempio aggiunge tag al sistema di bilanciamento del carico specificato.

Comando:

```
aws elb add-tags --load-balancer-name my-load-balancer --  
tags "Key=project,Value=Lima" "Key=department,Value=digital-media"
```

- Per API i dettagli, vedere [AddTags](#) in AWS CLI Command Reference.

### **apply-security-groups-to-load-balancer**

Il seguente esempio di codice mostra come utilizzare `apply-security-groups-to-load-balancer`.

AWS CLI

Associare un gruppo di sicurezza a un sistema di bilanciamento del carico in un VPC

Questo esempio associa un gruppo di sicurezza al sistema di bilanciamento del carico specificato in a. VPC

Comando:

```
aws elb apply-security-groups-to-load-balancer --load-balancer-name my-load-balancer
--security-groups sg-fc448899
```

Output:

```
{
  "SecurityGroups": [
    "sg-fc448899"
  ]
}
```

- Per API i dettagli, vedere [ApplySecurityGroupsToLoadBalancer](#) in AWS CLI Command Reference.

## attach-load-balancer-to-subnets

Il seguente esempio di codice mostra come utilizzare `attach-load-balancer-to-subnets`.

AWS CLI

Per collegare sottoreti a un sistema di bilanciamento del carico

Questo esempio aggiunge la sottorete specificata all'insieme di sottoreti configurate per il sistema di bilanciamento del carico specificato.

Comando:

```
aws elb attach-load-balancer-to-subnets --load-balancer-name my-load-balancer --
subnets subnet-0ecac448
```

Output:

```
{
  "Subnets": [
    "subnet-15aaab61",
    "subnet-0ecac448"
  ]
}
```

- Per i API dettagli, vedere [AttachLoadBalancerToSubnets](#) in Command Reference.AWS CLI

## configure-health-check

Il seguente esempio di codice mostra come utilizzare `configure-health-check`.

### AWS CLI

Per specificare le impostazioni del controllo dello stato delle istanze di backend EC2

Questo esempio specifica le impostazioni del controllo dello stato utilizzate per valutare lo stato delle istanze di backend. EC2

Comando:

```
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
```

Output:

```
{
  "HealthCheck": {
    "HealthyThreshold": 2,
    "Interval": 30,
    "Target": "HTTP:80/png",
    "Timeout": 3,
    "UnhealthyThreshold": 2
  }
}
```

- Per API i dettagli, vedere [ConfigureHealthCheck](#) in AWS CLI Command Reference.

## create-app-cookie-stickiness-policy

Il seguente esempio di codice mostra come utilizzare `create-app-cookie-stickiness-policy`.

### AWS CLI

Per generare una politica di aderenza per il sistema di HTTPS bilanciamento del carico

Questo esempio genera una politica di persistenza che segue la durata delle sessioni permanenti del cookie generato dall'applicazione.

Comando:

```
aws elb create-app-cookie-stickiness-policy --load-balancer-name my-load-balancer --policy-name my-app-cookie-policy --cookie-name my-app-cookie
```

- Per API i dettagli, consulta Command Reference. [CreateAppCookieStickinessPolicy](#) AWS CLI

## **create-lb-cookie-stickiness-policy**

Il seguente esempio di codice mostra come utilizzare `create-lb-cookie-stickiness-policy`.

AWS CLI

Per generare una politica di aderenza basata sulla durata per il sistema di bilanciamento del carico HTTPS

Questo esempio genera una politica di persistenza con una durata delle sessioni permanenti controllata dal periodo di scadenza specificato.

Comando:

```
aws elb create-lb-cookie-stickiness-policy --load-balancer-name my-load-balancer --policy-name my-duration-cookie-policy --cookie-expiration-period 60
```

- Per i API dettagli, vedere [CreateLbCookieStickinessPolicy](#) in Command Reference. AWS CLI

## **create-load-balancer-listeners**

Il seguente esempio di codice mostra come utilizzare `create-load-balancer-listeners`.

AWS CLI

Per creare HTTP listener per un sistema di bilanciamento del carico

Questo esempio crea un listener per il sistema di bilanciamento del carico sulla porta 80 utilizzando il protocollo. HTTP

Comando:

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
```

Per creare HTTPS listener per un sistema di bilanciamento del carico

Questo esempio crea un listener per il sistema di bilanciamento del carico sulla porta 443 utilizzando il protocollo. HTTPS

Comando:

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --  
listeners "Protocol=HTTPS,LoadBalancerPort=443,InstanceProtocol=HTTP,InstancePort=80"
```

- Per API i dettagli, vedere [CreateLoadBalancerListeners](#) in AWS CLI Command Reference.

## create-load-balancer-policy

Il seguente esempio di codice mostra come utilizzare `create-load-balancer-policy`.

AWS CLI

Per creare una politica che abiliti il Proxy Protocol su un sistema di bilanciamento del carico

Questo esempio crea una policy che abilita il Proxy Protocol sul load balancer specificato.

Comando:

```
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
```

Per creare una politica di SSL negoziazione utilizzando la politica di sicurezza consigliata

Questo esempio crea una politica di SSL negoziazione per il sistema di bilanciamento del HTTPS carico specificato utilizzando la politica di sicurezza consigliata.

Comando:

```
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
```

Per creare una politica di SSL negoziazione utilizzando una politica di sicurezza personalizzata

Questo esempio crea una politica di SSL negoziazione per il sistema di bilanciamento HTTPS del carico utilizzando una politica di sicurezza personalizzata abilitando i protocolli e i codici.

Comando:

```
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
```

Per creare una politica a chiave pubblica

Questo esempio crea una politica a chiave pubblica.

Comando:

```
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=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQ+dS74kj//c6x7R0tusUaeQCTgIUkayttRDWchuqo1pHC1u+n5xxXnBBE2ejbb2WRsKIQ5rXEeixsjFpFsojpSQKkzhVGI6mJVZBJDVKSHmswnwLBdofLhzvllpovBPTHe+o4haAWvDBALJU0pkSI1FecPHcs2hwx14zHoXy1e2k36A64nXW43wtfx5qcVSIxtCE0jnYRg7RPvybaGfQ+v6Iaxb/+7J5kEvZhTFQId+bSiJImF1FSUT1W1xwzBZPUbcUkkXDj45vC2s3Z8E+Lk7a3uZhvsQHLZnrFuWjBWGWvZ/MhZYgEXAMPLE
```

Per creare una politica di autenticazione del server di backend

Questo esempio crea una politica di autenticazione del server di backend che abilita l'autenticazione sull'istanza di backend utilizzando una politica a chiave pubblica.

Comando:

```
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
```

- Per API i dettagli, consulta AWS CLI Command [CreateLoadBalancerPolicyReference](#).

## create-load-balancer

Il seguente esempio di codice mostra come utilizzare `create-load-balancer`.

### AWS CLI

Per creare un sistema di HTTP bilanciamento del carico

Questo esempio crea un sistema di bilanciamento del carico con un HTTP listener in un VPC

Comando:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" --subnets subnet-15aabb61 --security-groups sg-a61988c3
```

Output:

```
{  
  "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"  
}
```

Questo esempio crea un sistema di bilanciamento del carico con un HTTP listener in -Classic.  
EC2

Comando:

```
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
```

Output:

```
{  
  "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"  
}
```

Per creare un sistema di bilanciamento del carico HTTPS

Questo esempio crea un sistema di bilanciamento del carico con un HTTPS listener in un VPC

Comando:

```
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-15aab61 --security-groups sg-a61988c3
```

Output:

```
{  
  "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"  
}
```

Questo esempio crea un sistema di bilanciamento del carico con un HTTPS listener in -Classic.  
EC2

Comando:

```
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
```

Output:

```
{  
  "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"  
}
```

Per creare un sistema di bilanciamento del carico interno

Questo esempio crea un sistema di bilanciamento del carico interno con un HTTP listener in un VPC

Comando:

```
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
```



**Output:**

```
{
  "DNSName": "internal-my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

- Per API i dettagli, vedere [CreateLoadBalancer](#) in AWS CLI Command Reference.

**delete-load-balancer-listeners**

Il seguente esempio di codice mostra come utilizzare `delete-load-balancer-listeners`.

**AWS CLI**

Per eliminare un listener dal sistema di bilanciamento del carico

Questo esempio elimina il listener per la porta specificata dal sistema di bilanciamento del carico specificato.

Comando:

```
aws elb delete-load-balancer-listeners --load-balancer-name my-load-balancer --load-balancer-ports 80
```

- Per API i dettagli, vedete [DeleteLoadBalancerListeners](#) in AWS CLI Command Reference.

**delete-load-balancer-policy**

Il seguente esempio di codice mostra come utilizzare `delete-load-balancer-policy`.

**AWS CLI**

Per eliminare una policy dal sistema di bilanciamento del carico

Questo esempio elimina la politica specificata dal sistema di bilanciamento del carico specificato. La policy non deve essere abilitata su nessun listener.

Comando:

```
aws elb delete-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-duration-cookie-policy
```

- Per API i dettagli, vedere [DeleteLoadBalancerPolicy](#) in AWS CLI Command Reference.

## **delete-load-balancer**

Il seguente esempio di codice mostra come utilizzare `delete-load-balancer`.

### AWS CLI

Per eliminare un sistema di bilanciamento del carico

Questo esempio elimina il sistema di bilanciamento del carico specificato.

Comando:

```
aws elb delete-load-balancer --load-balancer-name my-load-balancer
```

- Per API i dettagli, vedere [DeleteLoadBalancer](#) in AWS CLI Command Reference.

## **deregister-instances-from-load-balancer**

Il seguente esempio di codice mostra come utilizzare `deregister-instances-from-load-balancer`.

### AWS CLI

Per annullare la registrazione delle istanze da un sistema di bilanciamento del carico

Questo esempio annulla la registrazione dell'istanza specificata dal sistema di bilanciamento del carico specificato.

Comando:

```
aws elb deregister-instances-from-load-balancer --load-balancer-name my-load-balancer --instances i-d6f6fae3
```

Output:

```
{
  "Instances": [
    {
```

```
    "InstanceId": "i-207d9717"
  },
  {
    "InstanceId": "i-afefb49b"
  }
]
}
```

- Per i API dettagli, vedere [DeregisterInstancesFromLoadBalancer](#) in Command Reference.AWS CLI

## describe-account-limits

Il seguente esempio di codice mostra come utilizzare `describe-account-limits`.

AWS CLI

Per descrivere i limiti del Classic Load Balancer

L'`describe-account-limits` esempio seguente mostra i dettagli sui limiti del Classic Load Balancer per il tuo AWS account.

```
aws elb describe-account-limits
```

Output:

```
{
  "Limits": [
    {
      "Name": "classic-load-balancers",
      "Max": "20"
    },
    {
      "Name": "classic-listeners",
      "Max": "100"
    },
    {
      "Name": "classic-registered-instances",
      "Max": "1000"
    }
  ]
}
```

```
}
```

- Per API i dettagli, vedere [DescribeAccountLimits](#) in AWS CLI Command Reference.

## describe-instance-health

Il seguente esempio di codice mostra come utilizzare `describe-instance-health`.

### AWS CLI

Per descrivere lo stato delle istanze di un sistema di bilanciamento del carico

Questo esempio descrive lo stato delle istanze per il sistema di bilanciamento del carico specificato.

Comando:

```
aws elb describe-instance-health --load-balancer-name my-load-balancer
```

Output:

```
{
  "InstanceStates": [
    {
      "InstanceId": "i-207d9717",
      "ReasonCode": "N/A",
      "State": "InService",
      "Description": "N/A"
    },
    {
      "InstanceId": "i-afefb49b",
      "ReasonCode": "N/A",
      "State": "InService",
      "Description": "N/A"
    }
  ]
}
```

Descrivere lo stato di un'istanza per un sistema di bilanciamento del carico

Questo esempio descrive lo stato dell'istanza specificata per il sistema di bilanciamento del carico specificato.

Comando:

```
aws elb describe-instance-health --load-balancer-name my-load-balancer --  
instances i-7299c809
```

Di seguito è riportato un esempio di risposta per un'istanza in fase di registrazione.

Output:

```
{  
  "InstanceStates": [  
    {  
      "InstanceId": "i-7299c809",  
      "ReasonCode": "ELB",  
      "State": "OutOfService",  
      "Description": "Instance registration is still in progress."  
    }  
  ]  
}
```

Di seguito è riportato un esempio di risposta per un'istanza non integra.

Output:

```
{  
  "InstanceStates": [  
    {  
      "InstanceId": "i-7299c809",  
      "ReasonCode": "Instance",  
      "State": "OutOfService",  
      "Description": "Instance has failed at least the UnhealthyThreshold number  
of health checks consecutively."  
    }  
  ]  
}
```

- Per API i dettagli, vedere [DescribeInstanceHealth](#) in AWS CLI Command Reference.

## **describe-load-balancer-attributes**

Il seguente esempio di codice mostra come utilizzare `describe-load-balancer-attributes`.

## AWS CLI

Per descrivere gli attributi di un sistema di bilanciamento del carico

Questo esempio descrive gli attributi del load balancer specificato.

Comando:

```
aws elb describe-load-balancer-attributes --load-balancer-name my-load-balancer
```

Output:

```
{
  "LoadBalancerAttributes": {
    "ConnectionDraining": {
      "Enabled": false,
      "Timeout": 300
    },
    "CrossZoneLoadBalancing": {
      "Enabled": true
    },
    "ConnectionSettings": {
      "IdleTimeout": 30
    },
    "AccessLog": {
      "Enabled": false
    }
  }
}
```

- Per API i dettagli, vedere [DescribeLoadBalancerAttributes](#) in AWS CLI Command Reference.

## **describe-load-balancer-policies**

Il seguente esempio di codice mostra come utilizzare `describe-load-balancer-policies`.

## AWS CLI

Per descrivere tutte le politiche associate a un sistema di bilanciamento del carico

Questo esempio descrive tutte le politiche associate al sistema di bilanciamento del carico specificato.

## Comando:

```
aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer
```

## Output:

```
{
  "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"
    },
    .
    .
    .
  ]
}
```

Per descrivere una politica specifica associata a un sistema di bilanciamento del carico

Questo esempio descrive la politica specificata associata al sistema di bilanciamento del carico specificato.

Comando:

```
aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer --  
policy-name my-authentication-policy
```

Output:

```
{  
  "PolicyDescriptions": [  
    {  
      "PolicyAttributeDescriptions": [  
        {  
          "AttributeName": "PublicKeyPolicyName",  
          "AttributeValue": "my-PublicKey-policy"  
        }  
      ],  
      "PolicyName": "my-authentication-policy",  
      "PolicyTypeName": "BackendServerAuthenticationPolicyType"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [DescribeLoadBalancerPolicies](#) in AWS CLI Command Reference.

## describe-load-balancer-policy-types

Il seguente esempio di codice mostra come utilizzare `describe-load-balancer-policy-types`.

AWS CLI

Per descrivere i tipi di policy di load balancer definiti da Elastic Load Balancing

Questo esempio descrive i tipi di policy di load balancer che è possibile utilizzare per creare configurazioni di policy per il load balancer.

Comando:



**aws elb describe-load-balancer-policy-types**

Output:

```
{
  "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."
  }
]
}

```

- Per API i dettagli, vedere [DescribeLoadBalancerPolicyTypes](#) in AWS CLI Command Reference.

## describe-load-balancers

Il seguente esempio di codice mostra come utilizzare `describe-load-balancers`.

### AWS CLI

Per descrivere i sistemi di bilanciamento del carico

Questo esempio descrive tutti i tuoi sistemi di bilanciamento del carico.

Comando:

```
aws elb describe-load-balancers
```

Per descrivere uno dei tuoi sistemi di bilanciamento del carico

Questo esempio descrive il sistema di bilanciamento del carico specificato.

Comando:

```
aws elb describe-load-balancers --load-balancer-name my-load-balancer
```

Il seguente esempio di risposta si riferisce a un sistema di HTTPS bilanciamento del carico in a VPC

Output:

```
{
  "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"
}
}
]
```

- Per API i dettagli, vedere [DescribeLoadBalancers](#) in AWS CLI Command Reference.

## describe-tags

Il seguente esempio di codice mostra come utilizzare `describe-tags`.

### AWS CLI

Per descrivere i tag assegnati a un sistema di bilanciamento del carico

Questo esempio descrive i tag assegnati al sistema di bilanciamento del carico specificato.

Comando:

```
aws elb describe-tags --load-balancer-name my-load-balancer
```

Output:

```
{
  "TagDescriptions": [
    {
      "Tags": [
        {
          "Value": "lima",
          "Key": "project"
        },
        {
          "Value": "digital-media",
          "Key": "department"
        }
      ],
      "LoadBalancerName": "my-load-balancer"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeTags](#) in AWS CLI Command Reference.

## **detach-load-balancer-from-subnets**

Il seguente esempio di codice mostra come utilizzare `detach-load-balancer-from-subnets`.

AWS CLI

Per scollegare i sistemi di bilanciamento del carico dalle sottoreti

Questo esempio scollega il sistema di bilanciamento del carico specificato dalla sottorete specificata.

Comando:

```
aws elb detach-load-balancer-from-subnets --load-balancer-name my-load-balancer --  
subnets subnet-0ecac448
```

Output:

```
{  
  "Subnets": [  
    "subnet-15aaab61"  
  ]  
}
```

- Per API i dettagli, vedere [DetachLoadBalancerFromSubnets](#) in AWS CLI Command Reference.

## **disable-availability-zones-for-load-balancer**

Il seguente esempio di codice mostra come utilizzare `disable-availability-zones-for-load-balancer`.

AWS CLI

Per disabilitare le zone di disponibilità per un sistema di bilanciamento del carico

Questo esempio rimuove la zona di disponibilità specificata dal set di zone di disponibilità per il sistema di bilanciamento del carico specificato.

Comando:

```
aws elb disable-availability-zones-for-load-balancer --load-balancer-name my-load-  
balancer --availability-zones us-west-2a
```

Output:

```
{  
  "AvailabilityZones": [  
    "us-west-2b"  
  ]  
}
```

- Per API i dettagli, vedere [DisableAvailabilityZonesForLoadBalancer](#) in AWS CLI Command Reference.

## **enable-availability-zones-for-load-balancer**

Il seguente esempio di codice mostra come utilizzare `enable-availability-zones-for-load-balancer`.

### AWS CLI

Per abilitare le zone di disponibilità per un sistema di bilanciamento del carico

Questo esempio aggiunge la zona di disponibilità specificata al sistema di bilanciamento del carico specificato.

Comando:

```
aws elb enable-availability-zones-for-load-balancer --load-balancer-name my-Load-balancer --availability-zones us-west-2b
```

Output:

```
{
  "AvailabilityZones": [
    "us-west-2a",
    "us-west-2b"
  ]
}
```

- Per API i dettagli, vedere [EnableAvailabilityZonesForLoadBalancer](#) in AWS CLI Command Reference.

## **modify-load-balancer-attributes**

Il seguente esempio di codice mostra come utilizzare `modify-load-balancer-attributes`.

### AWS CLI

Per modificare gli attributi di un sistema di bilanciamento del carico

Questo esempio modifica l'`CrossZoneLoadBalancing` attributo del load balancer specificato.

Comando:



```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer --load-balancer-attributes "{\"CrossZoneLoadBalancing\":{\"Enabled\":true}}"
```

Output:

```
{
  "LoadBalancerAttributes": {
    "CrossZoneLoadBalancing": {
      "Enabled": true
    }
  },
  "LoadBalancerName": "my-load-balancer"
}
```

Questo esempio modifica l'ConnectionDraining attributo del load balancer specificato.

Comando:

```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer --load-balancer-attributes "{\"ConnectionDraining\":{\"Enabled\":true,\"Timeout\":300}}"
```

Output:

```
{
  "LoadBalancerAttributes": {
    "ConnectionDraining": {
      "Enabled": true,
      "Timeout": 300
    }
  },
  "LoadBalancerName": "my-load-balancer"
}
```

- Per API i dettagli, vedere [ModifyLoadBalancerAttributes](#) in AWS CLI Command Reference.

## register-instances-with-load-balancer

Il seguente esempio di codice mostra come utilizzare `register-instances-with-load-balancer`.

## AWS CLI

Per registrare istanze con un sistema di bilanciamento del carico

Questo esempio registra l'istanza specificata con il sistema di bilanciamento del carico specificato.

Comando:

```
aws elb register-instances-with-load-balancer --load-balancer-name my-load-balancer
--instances i-d6f6fae3
```

Output:

```
{
  "Instances": [
    {
      "InstanceId": "i-d6f6fae3"
    },
    {
      "InstanceId": "i-207d9717"
    },
    {
      "InstanceId": "i-afefb49b"
    }
  ]
}
```

- Per API i dettagli, vedere [RegisterInstancesWithLoadBalancer](#) in AWS CLI Command Reference.

## remove-tags

Il seguente esempio di codice mostra come utilizzare `remove-tags`.

## AWS CLI

Per rimuovere i tag da un sistema di bilanciamento del carico

Questo esempio rimuove un tag dal sistema di bilanciamento del carico specificato.

Comando:

```
aws elb remove-tags --load-balancer-name my-load-balancer --tags project
```

- Per API i dettagli, vedere [RemoveTags](#) in AWS CLI Command Reference.

## set-load-balancer-listener-ssl-certificate

Il seguente esempio di codice mostra come utilizzare `set-load-balancer-listener-ssl-certificate`.

### AWS CLI

Per aggiornare il SSL certificato per un sistema di HTTPS bilanciamento del carico

Questo esempio sostituisce il SSL certificato esistente per il sistema di bilanciamento del HTTPS carico specificato.

Comando:

```
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
```

- Per API i dettagli, vedere [SetLoadBalancerListenerSslCertificate](#) in AWS CLI Command Reference.

## set-load-balancer-policies-for-backend-server

Il seguente esempio di codice mostra come utilizzare `set-load-balancer-policies-for-backend-server`.

### AWS CLI

Per sostituire le politiche associate a una porta per un'istanza di backend

Questo esempio sostituisce le politiche attualmente associate alla porta specificata.

Comando:

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-balancer --instance-port 80 --policy-names my-ProxyProtocol-policy
```

Per rimuovere tutte le politiche attualmente associate a una porta sull'istanza di backend

Questo esempio rimuove tutte le politiche associate alla porta specificata.

Comando:

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-balancer --instance-port 80 --policy-names []
```

Per confermare che le politiche sono state rimosse, utilizzare il `describe-load-balancer-policies` comando.

- Per API i dettagli, vedere [SetLoadBalancerPoliciesForBackendServer](#) in AWS CLI Command Reference.

## set-load-balancer-policies-of-listener

Il seguente esempio di codice mostra come utilizzare `set-load-balancer-policies-of-listener`.

AWS CLI

Per sostituire le politiche associate a un listener

Questo esempio sostituisce le politiche attualmente associate al listener specificato.

Comando:

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer --load-balancer-port 443 --policy-names my-SSLNegotiation-policy
```

Per rimuovere tutte le politiche associate al tuo listener

Questo esempio rimuove tutte le politiche attualmente associate al listener specificato.

Comando:

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer --load-balancer-port 443 --policy-names []
```

Per confermare che le policy vengono rimosse dal load balancer, utilizzate il `describe-load-balancer-policies` comando.

- Per API i dettagli, vedere [SetLoadBalancerPoliciesOfListener](#) in AWS CLI Command Reference.

## Elastic Load Balancing - Esempi di utilizzo della versione 2 AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Elastic Load Balancing - Versione 2.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **add-listener-certificates**

Il seguente esempio di codice mostra come utilizzare `add-listener-certificates`.

AWS CLI

Per aggiungere un certificato a un listener sicuro

Questo esempio aggiunge il certificato specificato al listener sicuro specificato.

Comando:

```
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
```

Output:

```
{  
  "Certificates": [  
    {  
      "CertificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705"  
    }  
  ]  
}
```

```
{
  "CertificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705",
  "IsDefault": false
}
]
```

- Per API i dettagli, vedere [AddListenerCertificates](#) in AWS CLI Command Reference.

## add-tags

Il seguente esempio di codice mostra come utilizzare `add-tags`.

### AWS CLI

Per aggiungere tag a un sistema di bilanciamento del carico

L'`add-tags` esempio seguente aggiunge i department tag project and al sistema di bilanciamento del carico specificato.

```
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"
```

- Per API i dettagli, vedere [AddTags](#) in AWS CLI Command Reference.

## create-listener

Il seguente esempio di codice mostra come utilizzare `create-listener`.

### AWS CLI

Esempio 1: creare un HTTP ascoltatore

L'`create-listener` esempio seguente crea un HTTP listener per l'Application Load Balancer specificato che inoltra le richieste al gruppo target specificato.

```
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
```

Per ulteriori informazioni, vedere [Tutorial: Create an Application Load Balancer utilizzando la AWS CLI](#) guida utente per Application Load Balancers.

Esempio 2: Per creare un listener HTTPS

L'`create-listener` esempio seguente crea un HTTPS listener per l'Application Load Balancer specificato che inoltra le richieste al gruppo target specificato. È necessario specificare un SSL certificato per un listener. HTTPS È possibile creare e gestire certificati utilizzando AWS Certificate Manager (ACM). In alternativa, puoi creare un certificato utilizzando SSL/TLS tools, far firmare il certificato da un'autorità di certificazione (CA) e caricarlo su AWS Identity and Access Management (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 \  
  --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
```

Per ulteriori informazioni, consulta [Aggiungere un HTTPS listener](#) nella Guida utente per Application Load Balancers.

Esempio 3: creare un listener TCP

L'`create-listener` esempio seguente crea un TCP listener per il Network Load Balancer specificato che inoltra le richieste al gruppo target specificato.

```
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-targets/73e2d6bc24d8a067
```

```
--default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-tcp-targets/b6bba954d1361c78
```

Per ulteriori informazioni, vedere [Tutorial: Create a Network Load Balancer utilizzando la AWS CLI](#) guida utente per Network Load Balancer.

Esempio 4: Per creare un listener TLS

L'`create-listener` seguente crea un TLS listener per il Network Load Balancer specificato che inoltra le richieste al gruppo target specificato. È necessario specificare un SSL certificato per un listener. 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
```

Per ulteriori informazioni, consulta i [TLSlistener per il tuo Network Load Balancer](#) nella User Guide for Network Load Balancer.

Esempio 5: creare un listener UDP

L'`create-listener` seguente crea un UDP listener per il Network Load Balancer specificato che inoltra le richieste al gruppo target specificato.

```
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
```

Per ulteriori informazioni, vedere [Tutorial: Create a Network Load Balancer utilizzando la AWS CLI](#) guida utente per Network Load Balancer.



Esempio 6: Per creare un listener per il gateway e l'inoltro specificati

L'create-listener esempio seguente crea un listener per il Gateway Load Balancer specificato che inoltra le richieste al gruppo target specificato.

```
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
```

Output:

```
{
  "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"
              }
            ]
          }
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, vedere [Guida introduttiva all'utilizzo di Gateway Load Balancers AWS CLI nella Guida per l'utente di Gateway Load Balancers](#).

- Per i API dettagli, vedere [CreateListener](#) in Command Reference.AWS CLI

## create-load-balancer

Il seguente esempio di codice mostra come utilizzare `create-load-balancer`.

### AWS CLI

Esempio 1: creare un sistema di bilanciamento del carico connesso a Internet

L'`create-load-balancer` esempio seguente crea un Application Load Balancer con accesso a Internet e abilita le zone di disponibilità per le sottoreti specificate.

```
aws elbv2 create-load-balancer \  
  --name my-load-balancer \  
  --subnets subnet-b7d581c0 subnet-8360a9e7
```

Output:

```
{  
  "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"
  }
]
}
```

Per ulteriori informazioni, vedere [Tutorial: Create an Application Load Balancer utilizzando la AWS CLI](#) guida utente per Application Load Balancers.

Esempio 2: creare un load balancer interno

L'`create-load-balancer` esempio seguente crea un Application Load Balancer interno e abilita le zone di disponibilità per le sottoreti specificate.

```
aws elbv2 create-load-balancer \
  --name my-internal-load-balancer \
  --scheme internal \
  --subnets subnet-b7d581c0 subnet-8360a9e7
```

Output:

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, vedere [Tutorial: Create an Application Load Balancer utilizzando la AWS CLI](#) guida utente per Application Load Balancers.

### Esempio 3: creare un Network Load Balancer

L'`create-load-balancer` esempio seguente crea un Network Load Balancer con accesso a Internet e abilita la zona di disponibilità per la sottorete specificata. Utilizza una mappatura di sottorete per associare l'indirizzo IP elastico specificato all'interfaccia di rete utilizzata dai nodi di bilanciamento del carico per la zona di disponibilità.

```

aws elbv2 create-load-balancer \
  --name my-network-load-balancer \
  --type network \
  --subnet-mappings SubnetId=subnet-b7d581c0,AllocationId=eipalloc-64d5890a

```

Output:

```

{
  "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"
    }
]
}

```

Per ulteriori informazioni, vedere [Tutorial: Create a Network Load Balancer utilizzando la AWS CLI](#) guida utente per Network Load Balancer.

#### Esempio 4: creare un Gateway Load Balancer

L'create-load-balanceresempio seguente crea un Gateway Load Balancer e abilita le zone di disponibilità per le sottoreti specificate.

```

aws elbv2 create-load-balancer \
  --name my-gateway-load-balancer \
  --type gateway \
  --subnets subnet-dc83f691 subnet-a62583f9

```

Output:

```

{
  "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"
  }
]
}

```

Per ulteriori informazioni, vedere [Guida introduttiva all'utilizzo di Gateway Load Balancers AWS CLI nella Guida per l'utente di Gateway Load Balancers](#).

- Per i API dettagli, vedere [CreateLoadBalancer](#) in Command Reference.AWS CLI

## create-rule

Il seguente esempio di codice mostra come utilizzare `create-rule`.

### AWS CLI

Esempio 1: creare una regola utilizzando una condizione di percorso e un'azione di inoltra

L'`create-rule` esempio seguente crea una regola che inoltra le richieste al gruppo target specificato se URL contiene il modello specificato.

```

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
```

Contenuto di conditions-pattern.json.

```
[
  {
    "Field": "path-pattern",
    "PathPatternConfig": {
      "Values": ["/images/*"]
    }
  }
]
```

Esempio 2: creare una regola utilizzando una condizione host e una risposta fissa

L'create-ruleesempio seguente crea una regola che fornisce una risposta fissa se il nome host nell'intestazione dell'host corrisponde al nome host specificato.

```
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
```

Contenuto di conditions-host.json

```
[
  {
    "Field": "host-header",
    "HostHeaderConfig": {
      "Values": [ "*.example.com" ]
    }
  }
]
```

Contenuto di actions-fixed-response.json

```
[
```

```

    {
      "Type": "fixed-response",
      "FixedResponseConfig": {
        "MessageBody": "Hello world",
        "StatusCode": "200",
        "ContentType": "text/plain"
      }
    }
  ]

```

Esempio 3: creare una regola utilizzando una condizione di indirizzo IP di origine, un'azione di autenticazione e un'azione di inoltra

L'`create-rule` seguente crea una regola che autentica l'utente se l'indirizzo IP di origine corrisponde all'indirizzo IP specificato e inoltra la richiesta al gruppo di destinazione specificato se l'autenticazione ha esito positivo.

```

aws elbv2 create-rule \
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
  --priority 20 \
  --conditions file://conditions-source-ip.json \
  --actions file://actions-authenticate.json

```

Contenuto di `conditions-source-ip.json`

```

[
  {
    "Field": "source-ip",
    "SourceIpConfig": {
      "Values": ["192.0.2.0/24", "198.51.100.10/32"]
    }
  }
]

```

Contenuto di `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
}
]

```

- Per API i dettagli, vedere [CreateRule](#) in AWS CLI Command Reference.

## create-target-group

Il seguente esempio di codice mostra come utilizzare `create-target-group`.

### AWS CLI

Esempio 1: creare un gruppo target per un Application Load Balancer

L'esempio seguente crea un gruppo target per un Application Load Balancer in cui si registrano le destinazioni per ID di istanza (il tipo di destinazione è `instance`). Questo gruppo target utilizza il HTTP protocollo, la porta 80 e le impostazioni predefinite per il controllo dello stato di salute per un gruppo HTTP target.

```

aws elbv2 create-target-group \
  --name my-targets \
  --protocol HTTP \

```

```
--port 80 \  
--target-type instance \  
--vpc-id vpc-3ac0fb5f
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Create a target group](#) nella User Guide for Application Load Balancers.

Esempio 2: creare un gruppo target per indirizzare il traffico da un Application Load Balancer a una funzione Lambda

L'create-target-groupesempio seguente crea un gruppo target per un Application Load Balancer in cui la destinazione è una funzione Lambda (il tipo di destinazione è). Lambda Per impostazione predefinita, i controlli sanitari sono disabilitati per questo gruppo target.

```
aws elbv2 create-target-group \  
  --name my-lambda-target \  
  --target-type Lambda
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Lambda functions as targets](#) nella Guida per l'utente di Application Load Balancer.

Esempio 3: creare un gruppo target per un Network Load Balancer

L'create-target-groupesempio seguente crea un gruppo target per un Network Load Balancer in cui si registrano le destinazioni per indirizzo IP (il tipo di destinazione è ip). Questo gruppo target utilizza il TCP protocollo, la porta 80 e le impostazioni predefinite per il controllo dello stato di salute per un gruppo TCP target.

```
aws elbv2 create-target-group \  
  --name my-ip-targets \  
  --protocol TCP \  
  --port 80 \  
  --target-type ip
```

```
--target-type ip \  
--vpc-id vpc-3ac0fb5f
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Creare un gruppo target](#) nella Guida per l'utente di Network Load Balancers.

Esempio 4: creare un gruppo target per instradare il traffico da un Network Load Balancer a un Application Load Balancer

L'create-target-groupesempio seguente crea un gruppo target per un Network Load Balancer in cui si registra un Application Load Balancer come destinazione (il tipo di destinazione è). alb

```
aws elbv2 create-target-group --name --protocol --port 80 --target-type lab --vpc-id my-alb-target  
TCP vpc-3ac0fb5f
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Creare un gruppo target con un Application Load Balancer come destinazione nella User Guide for Network Load Balancers](#).

Esempio 5: creare un gruppo target per un Gateway Load Balancer

L'`create-target-group` seguente crea un gruppo target per un Gateway Load Balancer in cui la destinazione è un'istanza e il protocollo del gruppo di destinazione è GENEVE.

```
aws elbv2 create-target-group \
  --name my-glb-targetgroup \
  --protocol GENEVE \
  --port 6081 \
  --target-type instance \
  --vpc-id vpc-838475fe
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [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> > nella Gateway Load Balancer User Guide.

- Per i API AWS CLI dettagli, [CreateTargetGroup](#) consulta Command Reference.

## delete-listener

Il seguente esempio di codice mostra come utilizzare `delete-listener`.

### AWS CLI

Per eliminare un ascoltatore

L'`delete-listener` esempio seguente elimina il listener specificato.

```
aws elbv2 delete-listener \
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- Per API i dettagli, vedete [DeleteListener](#) in AWS CLI Command Reference.

## delete-load-balancer

Il seguente esempio di codice mostra come utilizzare `delete-load-balancer`.

### AWS CLI

Per eliminare un sistema di bilanciamento del carico

L'`delete-load-balancer` esempio seguente elimina il sistema di bilanciamento del carico specificato.

```
aws elbv2 delete-load-balancer \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

- Per API i dettagli, vedere [DeleteLoadBalancer](#) in AWS CLI Command Reference.

## delete-rule

Il seguente esempio di codice mostra come utilizzare `delete-rule`.

### AWS CLI

Per eliminare una regola

L'`delete-rule` esempio seguente elimina la regola specificata.

```
aws elbv2 delete-rule \  
  --rule-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/1291d13826f405c3
```

- Per API i dettagli, vedere [DeleteRule](#) in AWS CLI Command Reference.

## delete-target-group

Il seguente esempio di codice mostra come utilizzare `delete-target-group`.

### AWS CLI

Per eliminare un gruppo target

L'`delete-target-group` seguente elimina il gruppo target specificato.

```
aws elbv2 delete-target-group \  
  --target-group-arn arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminare un load balancer](#) nella Application Load Balancer Guide.

- Per API i dettagli, vedere [DeleteTargetGroup](#) in AWS CLI Command Reference.

## deregister-targets

Il seguente esempio di codice mostra come utilizzare `deregister-targets`.

### AWS CLI

Esempio 1: Annullare la registrazione di un obiettivo da un gruppo target

L'`deregister-targets` seguente rimuove l'istanza specificata dal gruppo target specificato.

```
aws elbv2 deregister-targets \  
  --target-group-arn arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \  
  --targets Id=i-1234567890abcdef0
```

Esempio 2: Annullare la registrazione di una destinazione registrata utilizzando le sostituzioni delle porte

L'`deregister-targets` seguente rimuove un'istanza da un gruppo di destinazione che è stata registrata utilizzando le sostituzioni di porta.

```
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
```

- Per API i dettagli, vedere [DeregisterTargets](#) in AWS CLI Command Reference.



## describe-account-limits

Il seguente esempio di codice mostra come utilizzare `describe-account-limits`.

### AWS CLI

Per descrivere i limiti dell'Elastic Load Balancing

L'esempio seguente mostra i limiti di Elastic Load Balancing per il tuo AWS account nella regione corrente.

```
aws elbv2 describe-account-limits
```

Output:

```
{
  "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"
  }
]
}
```

Per ulteriori informazioni, consulta [Quotas](#) nel riferimento AWS generale.

- Per API i dettagli, vedere [DescribeAccountLimits](#) in AWS CLI Command Reference.

## **describe-listener-certificates**

Il seguente esempio di codice mostra come utilizzare `describe-listener-certificates`.

### AWS CLI

Per descrivere i certificati per un listener sicuro

Questo esempio descrive i certificati per il listener sicuro specificato.

Comando:

```
aws elbv2 describe-listener-certificates --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

Output:

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeListenerCertificates](#) in AWS CLI Command Reference.

## describe-listeners

Il seguente esempio di codice mostra come utilizzare `describe-listeners`.

### AWS CLI

Per descrivere un ascoltatore

Questo esempio descrive l'ascoltatore specificato.

Comando:

```
aws elbv2 describe-listeners --listener-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

**Output:**

```
{
  "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"
    }
  ]
}
```

Per descrivere i listener di un sistema di bilanciamento del carico

Questo esempio descrive i listener per il sistema di bilanciamento del carico specificato.

Comando:

```
aws elbv2 describe-listeners --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

**Output:**

```
{
  "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"
  }
]
}

```

- Per API i dettagli, vedete [DescribeListeners](#) in AWS CLI Command Reference.

## describe-load-balancer-attributes

Il seguente esempio di codice mostra come utilizzare `describe-load-balancer-attributes`.

### AWS CLI

Per descrivere gli attributi del load balancer

L'output di esempio seguente visualizza gli attributi del load balancer specificato.

```
aws elbv2 describe-load-balancer-attributes \
  --load-balancer-arn arn:aws:elasticloadbalancing:us-
  west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

L'output di esempio seguente mostra gli attributi di un 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"
    }
  ]
}
```

L'output di esempio seguente include gli attributi per un 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"
  }
]
}
```

- Per API i dettagli, vedere [DescribeLoadBalancerAttributes](#) in AWS CLI Command Reference.

## describe-load-balancers

Il seguente esempio di codice mostra come utilizzare `describe-load-balancers`.

### AWS CLI

Per descrivere un sistema di bilanciamento del carico

Questo esempio descrive il load balancer specificato.

Comando:

```
aws elbv2 describe-load-balancers --load-balancer-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

Output:



```
{
  "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"
    }
  ]
}
```

Per descrivere tutti i sistemi di bilanciamento del carico

Questo esempio descrive tutti i sistemi di bilanciamento del carico.

Comando:

```
aws elbv2 describe-load-balancers
```

- Per API i dettagli, vedere [DescribeLoadBalancers](#) in AWS CLI Command Reference.

## describe-rules

Il seguente esempio di codice mostra come utilizzare `describe-rules`.

### AWS CLI

Esempio 1: descrivere una regola

L'`describe-rules` esempio seguente visualizza i dettagli della regola specificata.

```
aws elbv2 describe-rules \  
  --rule-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/9683b2d02a6cabee
```

Esempio 2: descrivere le regole per un ascoltatore

L'`describe-rules` esempio seguente visualizza i dettagli delle regole per il listener specificato. L'output include la regola predefinita e tutte le altre regole che hai aggiunto.

```
aws elbv2 describe-rules \  
  --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- Per API i dettagli, vedi [DescribeRules](#) in AWS CLI Command Reference.

## describe-ssl-policies

Il seguente esempio di codice mostra come utilizzare `describe-ssl-policies`.

### AWS CLI

Esempio 1: per elencare le politiche utilizzate per la SSL negoziazione in base al tipo di load balancer

L'`describe-ssl-policies` esempio seguente visualizza i nomi delle policy che è possibile utilizzare per la SSL negoziazione con un Application Load Balancer. L'esempio utilizza il `--query` parametro per visualizzare solo i nomi delle politiche.

```
aws elbv2 describe-ssl-policies \  
  --load-balancer-type application \  
  --query SslPolicies[*].Name
```

## Output:

```
[
  "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"
]
```

Esempio 2: per elencare le politiche che supportano un protocollo specifico

L'`describe-ssl-policies` seguente mostra i nomi delle policy che supportano il protocollo TLS 1.3. L'esempio utilizza il `--query` parametro per visualizzare solo i nomi delle politiche.

```
aws elbv2 describe-ssl-policies \
  --load-balancer-type application \
  --query SslPolicies[?contains(SslProtocols, 'TLSv1.3')].Name
```

## Output:

```
[
  "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"  
]
```

Esempio 3: Per visualizzare i codici di una politica

L'output di `aws elbv2 describe-ssl-policies` seguente visualizza i nomi dei codici per la politica specificata. L'esempio utilizza il `--query` parametro per visualizzare solo i nomi dei codici. La prima cifra dell'elenco ha priorità 1 e le altre cifre sono in ordine di priorità.

```
aws elbv2 describe-ssl-policies \  
  --names ELBSecurityPolicy-TLS13-1-2-2021-06 \  
  --query SslPolicies[*].Ciphers[*].Name
```

Output:

```
[  
  "TLS_AES_128_GCM_SHA256",  
  "TLS_AES_256_GCM_SHA384",  
  "TLS_CHACHA20_POLY1305_SHA256",  
  "ECDHE-ECDSA-AES128-GCM-SHA256",  
  "ECDHE-RSA-AES128-GCM-SHA256",  
  "ECDHE-ECDSA-AES128-SHA256",  
  "ECDHE-RSA-AES128-SHA256",  
  "ECDHE-ECDSA-AES256-GCM-SHA384",  
  "ECDHE-RSA-AES256-GCM-SHA384",  
  "ECDHE-ECDSA-AES256-SHA384",  
  "ECDHE-RSA-AES256-SHA384"  
]
```

Per ulteriori informazioni, consulta le [politiche di sicurezza](#) nella Guida dell'utente per Application Load Balancers.

- Per API i dettagli, vedere [DescribeSslPolicies](#) in AWS CLI Command Reference.

## describe-tags

Il seguente esempio di codice mostra come utilizzare `describe-tags`.

### AWS CLI

Per descrivere i tag assegnati a un sistema di bilanciamento del carico

Questo esempio descrive i tag assegnati al sistema di bilanciamento del carico specificato.

Comando:

```
aws elbv2 describe-tags --resource-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

Output:

```
{
  "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"
        }
      ]
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeTags](#) in AWS CLI Command Reference.

## describe-target-group-attributes

Il seguente esempio di codice mostra come utilizzare `describe-target-group-attributes`.

AWS CLI

Per descrivere gli attributi del gruppo target

L'`describe-target-group-attributes` esempio seguente visualizza gli attributi del gruppo target specificato.

```
aws elbv2 describe-target-group-attributes \
```

```
--target-group-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

L'output include gli attributi se il protocollo è HTTP o HTTPS e il tipo di destinazione è `instance`.

```
{
  "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"
    }
  ]
}
```

L'output seguente include gli attributi se il protocollo è HTTP or HTTPS e il tipo di destinazione è `lambda`.

```
{
  "Attributes": [
    {
      "Value": "false",
      "Key": "lambda.multi_value_headers.enabled"
    }
  ]
}
```

L'output seguente include gli attributi se il protocollo è TCPTLS,UDP, o TCP\_UDP.

```
{
  "Attributes": [
    {
      "Value": "false",
      "Key": "proxy_protocol_v2.enabled"
    },
    {
      "Value": "300",
      "Key": "deregistration_delay.timeout_seconds"
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeTargetGroupAttributes](#) in AWS CLI Command Reference.

## describe-target-groups

Il seguente esempio di codice mostra come utilizzare `describe-target-groups`.

### AWS CLI

Esempio 1: descrivere un gruppo target

L'`describe-target-groups` esempio seguente visualizza i dettagli per il gruppo target specificato.

```
aws elbv2 describe-target-groups \
  --target-group-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Output:

```
{
  "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"
    }
]
}

```

Esempio 2: Per descrivere tutti i gruppi target per un sistema di bilanciamento del carico

L'output di `aws elbv2 describe-target-groups` seguente mostra i dettagli di tutti i gruppi target per il sistema di bilanciamento del carico specificato. L'esempio utilizza il `--query` parametro per visualizzare solo i nomi dei gruppi target.

```

aws elbv2 describe-target-groups \
  --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
  --query TargetGroups[*].TargetGroupName

```

Output:

```

[
  "my-instance-targets",
  "my-ip-targets",
  "my-lambda-target"
]

```



Per ulteriori informazioni, consulta [Target groups](#) nella Application Load Balancers Guide.

- Per i API dettagli, consulta [DescribeTargetGroups](#) Command Reference.AWS CLI

## describe-target-health

Il seguente esempio di codice mostra come utilizzare `describe-target-health`.

### AWS CLI

Esempio 1: descrivere lo stato di salute degli obiettivi per un gruppo target

L'`describe-target-health` esempio seguente visualizza i dettagli sullo stato di salute per gli obiettivi del gruppo target specificato. Questi obiettivi sono salutari.

```
aws elbv2 describe-target-health \
  --target-group-arn arn:aws:elasticloadbalancing:us-
  west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Output:

```
{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-ceddcd4d",
        "Port": 80
      },
      "TargetHealth": {
        "State": "healthy"
      }
    },
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {
        "State": "healthy"
      }
    }
  ]
}
```

```

    }
  ]
}

```

Esempio 2: descrivere lo stato di salute di un bersaglio

L'output seguente visualizza i dettagli sullo stato di salute del bersaglio specificato. Questo bersaglio è sano.

```

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

```

Output:

```

{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {
        "State": "healthy"
      }
    }
  ]
}

```

L'output di esempio seguente è per un target il cui gruppo target non è specificato in un'azione per un ascoltatore. Questo target non può ricevere traffico dal sistema di bilanciamento del carico.

```

{
  "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"
    }
  ]
}
```

L'output di esempio seguente riguarda un target il cui gruppo target è stato appena specificato in un'azione per un listener. L'obiettivo è ancora in fase di registrazione.

```
{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {
        "State": "initial",
        "Reason": "Elb.RegistrationInProgress",
        "Description": "Target registration is in progress"
      }
    }
  ]
}
```

L'output di esempio seguente riguarda un obiettivo non integro.

```
{
  "TargetHealthDescriptions": [
    {
      "HealthCheckPort": "80",
      "Target": {
        "Id": "i-0f76fade",
        "Port": 80
      },
      "TargetHealth": {
        "State": "unhealthy",
```

```

        "Reason": "Target.Timeout",
        "Description": "Connection to target timed out"
      }
    ]
  }

```

L'output di esempio seguente riguarda un target che è una funzione Lambda e i controlli di integrità sono disabilitati.

```

{
  "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"
      }
    }
  ]
}

```

- Per API i dettagli, vedere [DescribeTargetHealth](#) in AWS CLI Command Reference.

## modify-listener

Il seguente esempio di codice mostra come utilizzare `modify-listener`.

### AWS CLI

Esempio 1: Per modificare l'azione predefinita in un'azione di inoltra

L'`modify-listener` esempio seguente modifica l'azione predefinita (in un'azione di inoltra) per il listener specificato.

```

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
```

Output:

```
{
  "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"
    }
  ]
}
```

Esempio 2: per modificare l'azione predefinita in un'azione di reindirizzamento

L'`modify-listener` esempio seguente modifica l'azione predefinita in un'azione di reindirizzamento per il listener specificato.

```
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
```

Output:

```
{
  "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"
  }
]
}

```

### Esempio 3: Per modificare il certificato del server

Questo esempio modifica il certificato del server per il HTTPS listener specificato.

```

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

```

Output:

```

{
  "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"
}
]
}

```

- Per API i dettagli, vedete [ModifyListener](#) in AWS CLI Command Reference.

## modify-load-balancer-attributes

Il seguente esempio di codice mostra come utilizzare `modify-load-balancer-attributes`.

### AWS CLI

Per abilitare la protezione da eliminazione

Questo esempio abilita la protezione dall'eliminazione per il sistema di bilanciamento del carico specificato.

Comando:

```

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

```

Output:

```

{
  "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"
  }
]
}
```

Per modificare il timeout di inattività

Questo esempio modifica il valore del timeout di inattività per il sistema di bilanciamento del carico specificato.

Comando:

```
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
```

Output:

```
{
  "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"
    }
  ]
}
```

Per abilitare i log di accesso

Questo esempio abilita i log di accesso per il sistema di bilanciamento del carico specificato. Tieni presente che il bucket S3 deve esistere nella stessa regione del load balancer e deve avere una policy allegata che consenta l'accesso al servizio Elastic Load Balancing.

Comando:

```
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
```

Output:

```
{
  "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"
  }
]
}

```

- Per i API dettagli, consulta Command Reference. [ModifyLoadBalancerAttributes](#) AWS CLI

## modify-rule

Il seguente esempio di codice mostra come utilizzare `modify-rule`.

### AWS CLI

Per modificare una regola

L'`modify-rule` esempio seguente aggiorna le azioni e le condizioni per la regola specificata.

```

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

```

Output:

```

{
  "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"
      }
    ]
  }
]
}

```

- Per API i dettagli, vedere [ModifyRule](#) in AWS CLI Command Reference.

## modify-target-group-attributes

Il seguente esempio di codice mostra come utilizzare `modify-target-group-attributes`.

### AWS CLI

Per modificare il timeout del ritardo di annullamento della registrazione

Questo esempio imposta il timeout del ritardo di annullamento della registrazione sul valore specificato per il gruppo target specificato.

Comando:

```

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

```

Output:

```

{
  "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"
  }
]
}

```

- Per i API dettagli, vedere [ModifyTargetGroupAttributes](#) in Command Reference.AWS CLI

## modify-target-group

Il seguente esempio di codice mostra come utilizzare `modify-target-group`.

### AWS CLI

Per modificare la configurazione del controllo dello stato di salute per un gruppo target

L'`modify-target-group` esempio seguente modifica la configurazione dei controlli di integrità utilizzati per valutare lo stato degli obiettivi per il gruppo target specificato. Si noti che, a causa del modo in cui CLI analizza le virgole, è necessario racchiudere l'intervallo relativo all'`--matcher` opzione tra virgolette singole anziché virgolette doppie.

```

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'

```

Output:

```
{
```

```

"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"
  }
]
}

```

Per ulteriori informazioni, consulta [Target groups](#) nella Application Load Balancers Guide.

- Per i API dettagli, consulta [ModifyTargetGroup](#) Command Reference.AWS CLI

## register-targets

Il seguente esempio di codice mostra come utilizzare `register-targets`.

### AWS CLI

Esempio 1: registrare le destinazioni con un gruppo target per ID di istanza

L'`register-targets` esempio seguente registra le istanze specificate con un gruppo target. Il gruppo target deve avere un tipo di destinazione di `instance`

```
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
```

Esempio 2: registrare le destinazioni con un gruppo di destinazione utilizzando le sostituzioni delle porte

L'`register-targets`esempio seguente registra l'istanza specificata con un gruppo di destinazione utilizzando più porte. Ciò consente di registrare i contenitori sulla stessa istanza delle destinazioni nel gruppo di destinazione.

```
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
```

Esempio 3: registrare le destinazioni con un gruppo target in base all'indirizzo IP

L'`register-targets`esempio seguente registra gli indirizzi IP specificati con un gruppo target. Il gruppo target deve avere un tipo di destinazione di `ip`.

```
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
```

Esempio 4: registrare una funzione Lambda come destinazione

L'`register-targets`esempio seguente registra gli indirizzi IP specificati con un gruppo di destinazione. Il gruppo target deve avere un tipo di destinazione di `lambda`. È necessario concedere l'autorizzazione a Elastic Load Balancing per richiamare la funzione Lambda.

```
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
```

- Per API i dettagli, consulta Command [RegisterTargets](#)Reference AWS CLI .

## remove-listener-certificates

Il seguente esempio di codice mostra come utilizzare `remove-listener-certificates`.

### AWS CLI

Per rimuovere un certificato da un listener sicuro

Questo esempio rimuove il certificato specificato dal listener sicuro specificato.

Comando:

```
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
```

- Per API i dettagli, vedere [RemoveListenerCertificates](#) in AWS CLI Command Reference.

## remove-tags

Il seguente esempio di codice mostra come utilizzare `remove-tags`.

### AWS CLI

Per rimuovere i tag da un sistema di bilanciamento del carico

L'esempio seguente rimuove i `department` tag `project` and dal sistema di bilanciamento del carico specificato.

```
aws elbv2 remove-tags \  
  --resource-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \  
  --tag-keys project department
```

- Per API i dettagli, vedere [RemoveTags](#) in AWS CLI Command Reference.

## set-ip-address-type

Il seguente esempio di codice mostra come utilizzare `set-ip-address-type`.

## AWS CLI

Per impostare il tipo di indirizzo di un sistema di bilanciamento del carico

Questo esempio imposta il tipo di indirizzo del sistema di bilanciamento del carico specificato su `dualstack`. Le sottoreti del sistema di bilanciamento del carico devono avere blocchi associati. IPv6 CIDR

Comando:

```
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
```

Output:

```
{
  "IpAddressType": "dualstack"
}
```

- Per API i dettagli, vedere [SetIpAddressType](#) in AWS CLI Command Reference.

## set-rule-priorities

Il seguente esempio di codice mostra come utilizzare `set-rule-priorities`.

### AWS CLI

Per impostare la priorità della regola

Questo esempio imposta la priorità della regola specificata.

Comando:

```
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
```

Output:

```
{
```



```

"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"
      }
    ]
  }
]
}

```

- Per API i dettagli, vedere [SetRulePriorities](#) in AWS CLI Command Reference.

## set-security-groups

Il seguente esempio di codice mostra come utilizzare `set-security-groups`.

### AWS CLI

Per associare un gruppo di sicurezza a un sistema di bilanciamento del carico

Questo esempio associa il gruppo di sicurezza specificato al sistema di bilanciamento del carico specificato.

Comando:

```

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

```

**Output:**

```
{
  "SecurityGroupIds": [
    "sg-5943793c"
  ]
}
```

- Per API i dettagli, vedere [SetSecurityGroups](#) in AWS CLI Command Reference.

**set-subnets**

Il seguente esempio di codice mostra come utilizzare `set-subnets`.

**AWS CLI**

Per abilitare le zone di disponibilità per un sistema di bilanciamento del carico

Questo esempio abilita la zona di disponibilità per la sottorete specificata per il sistema di bilanciamento del carico specificato.

Comando:

```
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
```

**Output:**

```
{
  "AvailabilityZones": [
    {
      "SubnetId": "subnet-8360a9e7",
      "ZoneName": "us-west-2a"
    },
    {
      "SubnetId": "subnet-b7d581c0",
      "ZoneName": "us-west-2b"
    }
  ]
}
```

- Per API i dettagli, vedere [SetSubnets](#) in AWS CLI Command Reference.

## Esempi di utilizzo di Elastic Transcoder AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Elastic Transcoder.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **cancel-job**

Il seguente esempio di codice mostra come utilizzare `cancel-job`.

#### AWS CLI

Annullare un lavoro per ElasticTranscoder

Questo annulla il lavoro specificato per ElasticTranscoder.

Comando:

```
aws elastictranscoder cancel-job --id 333333333333-abcde3
```

- Per API i dettagli, vedere [CancelJob](#) in AWS CLI Command Reference.

#### **create-job**

Il seguente esempio di codice mostra come utilizzare `create-job`.

## AWS CLI

Per creare un lavoro per ElasticTranscoder

L'create-jobesempio seguente crea un lavoro per 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
```

Contenuto di inputs.json.

```
[{  
  "Key": "ETS_example_file.mp4",  
  "FrameRate": "auto",  
  "Resolution": "auto",  
  "AspectRatio": "auto",  
  "Interlaced": "auto",  
  "Container": "mp4"  
}]
```

Contenuto di outputs.json:

```
[  
  {  
    "Key": "webm/ETS_example_file-kindlefirehd.webm",  
    "Rotate": "0",  
    "PresetId": "1351620000001-100250"  
  }  
]
```

Contenuto di user-metadata.json.

```
{  
  "Food type": "Italian",  
  "Cook book": "recipe notebook"  
}
```

## Output:

```
{
  "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": "3333333333333-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"
    }
}

```

- Per i API dettagli, vedere [CreateJob](#) in Command Reference.AWS CLI

## create-pipeline

Il seguente esempio di codice mostra come utilizzare `create-pipeline`.

### AWS CLI

Per creare una pipeline per ElasticTranscoder

L'`create-pipeline`esempio seguente crea una pipeline per. 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

```

Contenuto di `content-config.json`.

```

{
  "Bucket": "salesoffice.example.com-public-promos",
  "Permissions": [
    {
      "GranteeType": "Email",
      "Grantee": "marketing-promos@example.com",

```

```

        "Access":[
            "FullControl"
        ]
    },
    "StorageClass":"Standard"
}

```

Contenuto di `thumbnail-config.json`.

```

{
  "Bucket":"salesoffice.example.com-public-promos-thumbnails",
  "Permissions":[
    {
      "GranteeType":"Email",
      "Grantee":"marketing-promos@example.com",
      "Access":[
        "FullControl"
      ]
    }
  ],
  "StorageClass":"ReducedRedundancy"
}

```

Output:

```

{
  "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"
    }
  ]
}

```

- Per API i dettagli, vedere [CreatePipeline](#) in AWS CLI Command Reference.

## create-preset

Il seguente esempio di codice mostra come utilizzare `create-preset`.



## AWS CLI

Per creare un preset per ElasticTranscoder

L'create-presetesempio seguente crea un preset per 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
```

Contenuto di 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"
    }
]
}
```

### Contenuto di audio.json.

```
{
  "Codec": "AAC",
  "CodecOptions": {
    "Profile": "AAC-LC"
  },
  "SampleRate": "44100",
  "BitRate": "96",
  "Channels": "2"
}
```

### Contenuto di thumbnails.json.

```
{
  "Format": "png",
  "Interval": "120",
  "MaxWidth": "auto",
  "MaxHeight": "auto",
  "SizingPolicy": "Fit",
  "PaddingPolicy": "Pad"
}
```

### Output:

```
{
  "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": ""
}
```

- Per API i dettagli, vedere [CreatePreset](#) in AWS CLI Command Reference.

## delete-pipeline

Il seguente esempio di codice mostra come utilizzare `delete-pipeline`.

### AWS CLI

Per eliminare la ElasticTranscoder pipeline specificata

Ciò elimina la tubazione specificata ElasticTranscoder .

Comando:

```
aws elastictranscoder delete-pipeline --id 111111111111-abcde1
```

Output:

```
{
  "Success": "true"
}
```

- Per API i dettagli, vedere [DeletePipeline](#) in AWS CLI Command Reference.

## delete-preset

Il seguente esempio di codice mostra come utilizzare `delete-preset`.

## AWS CLI

Per eliminare la ElasticTranscoder preimpostazione specificata

Questo cancella la preimpostazione specificata ElasticTranscoder .

Comando:

```
aws elastictranscoder delete-preset --id 555555555555-abcde5
```

- Per API i dettagli, vedere [DeletePreset](#) in AWS CLI Command Reference.

## **list-jobs-by-pipeline**

Il seguente esempio di codice mostra come utilizzare `list-jobs-by-pipeline`.

### AWS CLI

Per recuperare un elenco di ElasticTranscoder lavori nella pipeline specificata

Questo esempio recupera un elenco di ElasticTranscoder lavori nella pipeline specificata.

Comando:

```
aws elastictranscoder list-jobs-by-pipeline --pipeline-id 111111111111-abcde1
```

Output:

```
{  
  "Jobs": []  
}
```

- Per API i dettagli, vedere [ListJobsByPipeline](#) in AWS CLI Command Reference.

## **list-jobs-by-status**

Il seguente esempio di codice mostra come utilizzare `list-jobs-by-status`.

### AWS CLI

Per recuperare un elenco di ElasticTranscoder lavori con lo stato Completo

Questo esempio recupera un elenco di ElasticTranscoder lavori con lo stato Completo.

Comando:

```
aws elastictranscoder list-jobs-by-status --status Complete
```

Output:

```
{
  "Jobs": []
}
```

- Per API i dettagli, vedere [ListJobsByStatus](#) in AWS CLI Command Reference.

## list-pipelines

Il seguente esempio di codice mostra come utilizzare `list-pipelines`.

AWS CLI

Per recuperare un elenco di pipeline ElasticTranscoder

Questo esempio recupera un elenco di tubazioni. ElasticTranscoder

Comando:

```
aws elastictranscoder list-pipelines
```

Output:

```
{
  "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/3333333333333-abcde1"
  }
]
}

```

- Per API i dettagli, vedere [ListPipelines](#) in AWS CLI Command Reference.

## list-presets

Il seguente esempio di codice mostra come utilizzare `list-presets`.

### AWS CLI

Per recuperare un elenco di ElasticTranscoder preimpostazioni

Questo esempio recupera un elenco di preimpostazioni. ElasticTranscoder

Comando:

```
aws elastictranscoder list-presets --max-items 2
```

Output:

```

{
  "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=="
}

```

- Per API i dettagli, consultate AWS CLI Command [ListPresetsReference](#).

## read-job

Il seguente esempio di codice mostra come utilizzare `read-job`.

### AWS CLI

Per recuperare un lavoro ElasticTranscoder

Questo esempio recupera il lavoro specificato ElasticTranscoder .

Comando:

```
aws elastictranscoder read-job --id 1533838012294-example
```

Output:

```
{
  "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": "3333333333333-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"
}
}

```

- Per API i dettagli, vedere [ReadJob](#) in AWS CLI Command Reference.

## read-pipeline

Il seguente esempio di codice mostra come utilizzare `read-pipeline`.

### AWS CLI

Per recuperare una pipeline ElasticTranscoder

Questo esempio recupera la pipeline specificata. ElasticTranscoder

Comando:

```
aws elastictranscoder read-pipeline --id 333333333333-abcde3
```

## Output:

```
{
  "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": "3333333333333-abcde3",
    "Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:pipeline/3333333333333-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"
  }
]
```

- Per API i dettagli, vedere [ReadPipeline](#) in AWS CLI Command Reference.

## read-preset

Il seguente esempio di codice mostra come utilizzare `read-preset`.

### AWS CLI

Per recuperare un ElasticTranscoder preset

Questo esempio recupera la preimpostazione specificata. ElasticTranscoder

Comando:

```
aws elastictranscoder read-preset --id 1351620000001-500020
```

Output:

```
{
  "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"
}
}

```

- Per API i dettagli, vedete [ReadPreset](#) in AWS CLI Command Reference.

## update-pipeline-notifications

Il seguente esempio di codice mostra come utilizzare `update-pipeline-notifications`.

### AWS CLI

Per aggiornare le notifiche di una ElasticTranscoder pipeline



Questo esempio aggiorna le notifiche della ElasticTranscoder pipeline specificata.

Comando:

```
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
```

Output:

```
{  
  "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"
}
}

```

- Per API i dettagli, vedere [UpdatePipelineNotifications](#) in AWS CLI Command Reference.

## update-pipeline-status

Il seguente esempio di codice mostra come utilizzare `update-pipeline-status`.

### AWS CLI

Per aggiornare lo stato di una ElasticTranscoder pipeline

Questo esempio aggiorna lo stato della ElasticTranscoder tubazione specificata.

Comando:

```
aws elastictranscoder update-pipeline-status --id 111111111111-abcde1 --
status Paused
```

Output:

```

{
  "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"
}
}

```

- Per API i dettagli, vedere [UpdatePipelineStatus](#) in AWS CLI Command Reference.

## update-pipeline

Il seguente esempio di codice mostra come utilizzare `update-pipeline`.

### AWS CLI

Per aggiornare una ElasticTranscoder pipeline

L'update-pipelineesempio seguente aggiorna la ElasticTranscoder pipeline specificata.

```
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
```

Contenuto di content-config.json.

```
{  
  "Bucket": "salesoffice.example.com-public-promos",  
  "Permissions": [  
    {  
      "GranteeType": "Email",  
      "Grantee": "marketing-promos@example.com",  
      "Access": [  
        "FullControl"  
      ]  
    }  
  ],  
  "StorageClass": "Standard"  
}
```

Contenuto di thumbnail-config.json.

```
{  
  "Bucket": "salesoffice.example.com-public-promos-thumbnails",  
  "Permissions": [  
    {  
      "GranteeType": "Email",  
      "Grantee": "marketing-promos@example.com",  
      "Access": [  
        "FullControl"  
      ]  
    }  
  ],  
  "StorageClass": "ReducedRedundancy"
```

```
}
```

**Output:**

```
{
  "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-1:1112223333444:ETS_Errors"
    },
    "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
    "InputBucket": "ets-example",
    "Id": "3333333333333-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"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [UpdatePipeline](#) in AWS CLI Command Reference.

## ElastiCache esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with ElastiCache.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **add-tags-to-resource**

Il seguente esempio di codice mostra come utilizzare `add-tags-to-resource`.

#### AWS CLI

Per aggiungere tag a una risorsa

L'add-tags-to-resources esempio seguente aggiunge fino a 10 tag, coppie chiave-valore, a una risorsa cluster o snapshot.

```
aws elasticache add-tags-to-resource \  
  --resource-name "arn:aws:elasticache:us-east-1:1234567890:cluster:my-mem-  
cluster" \  
  --tags '{"20150202":15, "ElastiCache":"Service"}'
```

Output:

```
{  
  "TagList": [  
    {  
      "Value": "20150202",  
      "Key": "APIVersion"  
    },  
    {  
      "Value": "ElastiCache",  
      "Key": "Service"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Monitoraggio dei costi con i tag di allocazione dei costi](#) nella Guida per l'utente di ElastiCache.

- Per API i dettagli, consulta Command [AddTagsToResource](#) Reference AWS CLI .

## authorize-cache-security-group-ingress

Il seguente esempio di codice mostra come utilizzare authorize-cache-security-group-ingress.

AWS CLI

Per autorizzare l'ingresso del gruppo di sicurezza della cache

L'authorize-cache-security-group-ingress esempio seguente consente l'accesso alla rete a un gruppo di sicurezza della cache.

```
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"
```

Il comando non produce alcun output.

Per ulteriori informazioni, consulta [Self-Service Updates in Amazon ElastiCache nella Elasticache User Guide](#).

- Per API i dettagli, consulta Command [AuthorizeCacheSecurityGroupIngress](#) Reference AWS CLI .

## batch-apply-update-action

Il seguente esempio di codice mostra come utilizzare `batch-apply-update-action`.

### AWS CLI

Per applicare un aggiornamento del servizio

L'`batch-apply-update-action` seguente applica un aggiornamento del servizio a un cluster Redis.

```
aws elasticache batch-apply-update-action \  
--service-update-name elc-xxxxx406-xxx \  
--replication-group-ids test-cluster
```

Output:

```
{  
  "ProcessedUpdateActions": [  
    {  
      "ReplicationGroupId": "pat-cluster",  
      "ServiceUpdateName": "elc-xxxxx406-xxx",  
      "UpdateActionStatus": "waiting-to-start"  
    }  
  ],  
  "UnprocessedUpdateActions": []  
}
```

Per ulteriori informazioni, consulta [Self-Service Updates in Amazon ElastiCache nella Elasticache User Guide](#).



- Per API i dettagli, consulta Command [BatchApplyUpdateAction](#)Reference AWS CLI .

## batch-stop-update-action

Il seguente esempio di codice mostra come utilizzare `batch-stop-update-action`.

### AWS CLI

Per interrompere un aggiornamento del servizio

L'`batch-stop-update-action`esempio seguente applica un aggiornamento del servizio a un cluster Redis.

```
aws elasticache batch-stop-update-action \  
  --service-update-name elc-xxxxx406-xxx \  
  --replication-group-ids test-cluster
```

Output:

```
{  
  "ProcessedUpdateActions": [  
    {  
      "ReplicationGroupId": "pat-cluster",  
      "ServiceUpdateName": "elc-xxxxx406-xxx",  
      "UpdateActionStatus": "stopping"  
    }  
  ],  
  "UnprocessedUpdateActions": []  
}
```

Per ulteriori informazioni, consulta [Self-Service Updates in Amazon ElastiCache nella Elasticache User Guide](#).

- Per API i dettagli, consulta Command [BatchStopUpdateAction](#)Reference AWS CLI .

## copy-snapshot

Il seguente esempio di codice mostra come utilizzare `copy-snapshot`.

### AWS CLI

Per copiare un'istantanea

L'copy-snapshotesempio seguente crea una copia di un'istantanea esistente.

```
aws elasticache copy-snapshot \  
  --source-snapshot-name "my-snapshot" \  
  --target-snapshot-name "my-snapshot-copy"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Esportazione di un backup nella Guida](#) per l'utente di ElastiCache.

- Per API i dettagli, consulta Command [CopySnapshotReference](#) AWS CLI .

## create-cache-cluster

Il seguente esempio di codice mostra come utilizzare `create-cache-cluster`.

### AWS CLI

Per creare un cluster di cache

L'esempio seguente crea un cluster di cache utilizzando il motore Redis.

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id "cluster-test" \  
  --engine redis \  
  --cache-node-type cache.m5.large \  
  --num-cache-nodes 1
```

Output:

```
{  
  "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
```

```
}  
}
```

Per ulteriori informazioni, consulta [Creazione di un cluster nella Guida](#) per l'utente di Elasticache.

- Per API i dettagli, consulta AWS CLI Command [CreateCacheCluster](#)Reference.

## **create-cache-parameter-group**

Il seguente esempio di codice mostra come utilizzare `create-cache-parameter-group`.

### AWS CLI

Per creare un gruppo di parametri di cache

L'`create-cache-parameter-group`esempio seguente crea un nuovo gruppo di parametri Amazon ElastiCache cache.

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-family "redis5.0" \  
  --cache-parameter-group-name "mygroup" \  
  --description "mygroup"
```

Output:

```
{  
  "CacheParameterGroup": {  
    "CacheParameterGroupName": "mygroup",  
    "CacheParameterGroupFamily": "redis5.0",  
    "Description": "my group"  
  }  
}
```

Per ulteriori informazioni, consulta [Creazione di un gruppo di parametri nella Guida](#) per l'utente di Elasticache.

- Per API i dettagli, consulta AWS CLI Command [CreateCacheParameterGroup](#)Reference.

## **create-cache-subnet-group**

Il seguente esempio di codice mostra come utilizzare `create-cache-subnet-group`.

## AWS CLI

Per creare un gruppo di sottoreti di cache

L'`create-cache-subnet-group` seguente crea un nuovo gruppo di sottoreti di cache.

```
aws elasticache create-cache-subnet-group \  
  --cache-subnet-group-name "mygroup" \  
  --cache-subnet-group-description "my subnet group" \  
  --subnet-ids "subnet-xxxxec4f"
```

Output:

```
{  
  "CacheSubnetGroup": {  
    "CacheSubnetGroupName": "mygroup",  
    "CacheSubnetGroupDescription": "my subnet group",  
    "VpcId": "vpc-a3e97cdb",  
    "Subnets": [  
      {  
        "SubnetIdentifier": "subnet-xxxxec4f",  
        "SubnetAvailabilityZone": {  
          "Name": "us-west-2d"  
        }  
      }  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Creazione di un gruppo di sottoreti di cache](#) nella Guida per l'utente di Elasticache.

- Per API i dettagli, consulta Command [CreateCacheSubnetGroup](#) Reference AWS CLI .

## `create-global-replication-group`

Il seguente esempio di codice mostra come utilizzare `create-global-replication-group`.

## AWS CLI

Per creare un gruppo di replica globale

L'create-global-replication-groupesempio seguente crea un nuovo gruppo di replica globale.

```
aws elasticache create-global-replication-group \  
  --global-replication-group-id-suffix my-global-replication-group \  
  --primary-replication-group-id my-primary-cluster
```

Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, consulta [Replica tra AWS regioni utilizzando Global Datastore nella Guida per l'utente di Elasticache](#).

- Per i API dettagli, consulta Command Reference. [CreateGlobalReplicationGroupAWS CLI](#)

## create-replication-group

Il seguente esempio di codice mostra come utilizzare `create-replication-group`.

### AWS CLI

Per creare un gruppo di replica

L'esempio seguente crea un gruppo di replica Redis (modalità cluster disabilitata) o Redis (modalità cluster abilitata). Questa operazione è valida solo per Redis.

```
aws elasticache create-replication-group \  
  --replication-group-id "mygroup" \  
  --replication-group-description "my group" \  
  --engine "redis" \  
  --cache-node-type "cache.m5.large"
```

Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, consulta [Creazione di un gruppo di replica Redis](#) nella Guida per l'utente di ElastiCache.

- Per i API dettagli, consulta [CreateReplicationGroup](#) Command Reference.AWS CLI

## create-snapshot

Il seguente esempio di codice mostra come utilizzare `create-snapshot`.

### AWS CLI

Per creare un'istantanea

L'esempio seguente crea un'istantanea utilizzando il motore Redis.

```
aws elasticache create-snapshot \  
  --snapshot-name mynsnapshot \  
  --cache-cluster-id cluster-test
```

Output:

```
{  
  "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"  
      }  
    ]  
  }  
}
```



```
}  
}
```

Per ulteriori informazioni, consulta [Backup and Restore ElastiCache for Redis](#) nella Guida per l'utente di ElastiCache.

- Per API i dettagli, consulta Command [CreateSnapshotReference AWS CLI](#) .

## create-user-group

Il seguente esempio di codice mostra come utilizzare `create-user-group`.

### AWS CLI

Per creare un gruppo di utenti

L'`create-user-group`esempio seguente crea un nuovo gruppo di utenti.

```
aws elasticache create-user-group \  
  --user-group-id myusergroup \  
  --engine redis \  
  --user-ids default
```

Output:

```
{  
  "UserGroupId": "myusergroup",  
  "Status": "creating",  
  "Engine": "redis",  
  "UserIds": [  
    "default"  
  ],  
  "ReplicationGroups": [],  
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxx52:usergroup:myusergroup"  
}
```

Per ulteriori informazioni, consulta [Authenticating Users with Role-Based Access Control \(RBAC\)](#) nella Guida per l'utente di ElastiCache.

- Per i API dettagli, consulta Command Reference. [CreateUserGroupAWS CLI](#)

## create-user

Il seguente esempio di codice mostra come utilizzare `create-user`.

### AWS CLI

Per creare un utente

L'esempio seguente crea un nuovo utente.

```
aws elasticache create-user \  
  --user-id user1 \  
  --user-name myUser \  
  --passwords mYnuUzrpAxXw2rdzx \  
  --engine redis \  
  --access-string "on ~app:* -@all +@read"
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Authenticating Users with Role-Based Access Control \(RBAC\)](#) nella Guida per l'utente di Elasticache.

- Per i API dettagli, consulta Command Reference. [CreateUser](#) AWS CLI

## decrease-node-groups-in-global-replication-group

Il seguente esempio di codice mostra come utilizzare `decrease-node-groups-in-global-replication-group`.

### AWS CLI

Per ridurre il numero di gruppi di nodi in un gruppo di replica globale

Quanto segue `decrease-node-groups-in-global-replication-group` riduce il numero dei gruppi di nodi utilizzando il motore Redis.

```
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
```

Output:

```
{  
  "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
}
}
```

Per ulteriori informazioni, consulta [Replica tra AWS regioni utilizzando Global Datastore nella Guida per l'utente di Elasticache](#).

- Per i API dettagli, consulta Command Reference.  
[DecreaseNodeGroupsInGlobalReplicationGroup](#) AWS CLI

## decrease-replica-count

Il seguente esempio di codice mostra come utilizzare `decrease-replica-count`.

### AWS CLI

Per ridurre il numero di repliche

L'`decrease-replica-count` esempio seguente riduce dinamicamente il numero di repliche in un gruppo di replica Redis (modalità cluster disabilitata) o il numero di nodi di replica in uno o più

gruppi di nodi (shard) di un gruppo di replica Redis (modalità cluster abilitata). Questa operazione viene eseguita senza tempi di inattività del cluster.

```
aws elasticache decrease-replica-count \  
  --replication-group-id my-cluster \  
  --apply-immediately \  
  --new-replica-count 2
```

Output:

```
{  
  "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
  }
}
```

Per ulteriori informazioni, consulta [Modifica del numero di repliche](#) nella Guida per l'utente di Elasticache.

- Per API i dettagli, consulta Command [DecreaseReplicaCount](#)Reference AWS CLI .

## delete-cache-cluster

Il seguente esempio di codice mostra come utilizzare `delete-cache-cluster`.

### AWS CLI

Per eliminare un cluster di cache

L'`delete-cache-cluster` esempio seguente elimina il cluster specificato precedentemente assegnato. Il comando elimina tutti i nodi di cache associati, gli endpoint dei nodi e il cluster stesso. Quando ricevi una risposta corretta da questa operazione, Amazon inizia ElastiCache immediatamente a eliminare il cluster; non puoi annullare o annullare questa operazione.

Questa operazione non è valida per quanto segue:

Cluster Redis (modalità cluster abilitata) Un cluster che è l'ultima replica di lettura di un gruppo di nodi di replica GroupA (shard) con modalità Multi-AZ abilitata Un cluster di replica da un cluster di gruppo A di replica Redis (abilitata in modalità cluster) che non si trova nello stato disponibile

```
aws elasticache delete-cache-cluster \
  --cache-cluster-id "my-cluster-002"
```

Output:

```
{
  "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-xxxxxxxxxx7b",
    "Status": "active"
  }
],
"ReplicationGroupId": "my-cluster",
"SnapshotRetentionLimit": 0,
"SnapshotWindow": "07:30-08:30",
"TransitEncryptionEnabled": false,
"AtRestEncryptionEnabled": false
}
}
```

[Per ulteriori informazioni, consulta Eliminazione di un cluster nella Guida per l'utente di ElastiCache.](#)

- Per API i dettagli, consulta Command [DeleteCacheCluster](#) Reference AWS CLI .



## delete-cache-parameter-group

Il seguente esempio di codice mostra come utilizzare `delete-cache-parameter-group`.

### AWS CLI

Per eliminare un gruppo di parametri della cache

L'esempio seguente elimina il gruppo di parametri di cache specificato. Non è possibile eliminare un gruppo di parametri di cache se è associato a qualsiasi cluster di cache.

```
aws elasticache delete-cache-parameter-group \  
  --cache-parameter-group-name myparamgroup
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di un gruppo di parametri](#) nella Guida per l'utente di Elasticache.

- Per API i dettagli, consulta Command [DeleteCacheParameterGroup](#) Reference AWS CLI .

## delete-cache-subnet-group

Il seguente esempio di codice mostra come utilizzare `delete-cache-subnet-group`.

### AWS CLI

Per eliminare un gruppo di sottoreti di cache

L'esempio seguente elimina il gruppo di sottoreti di cache specificato. Non è possibile eliminare un gruppo di sottoreti di cache se è associato a qualsiasi cluster.

```
aws elasticache delete-cache-subnet-group \  
  --cache-subnet-group-name "mygroup"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di un gruppo di sottoreti](#) nella Guida per l'utente di Elasticache.

- Per i API dettagli, consulta [DeleteCacheSubnetGroup](#) Command Reference. AWS CLI

## delete-global-replication-group

Il seguente esempio di codice mostra come utilizzare `delete-global-replication-group`.

### AWS CLI

Per eliminare un gruppo di replica globale

L'esempio seguente elimina un nuovo gruppo di replica globale.

```
aws elasticache delete-global-replication-group \  
  --global-replication-group-id my-global-replication-group \  
  --retain-primary-replication-group
```

Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, consulta [Replica tra AWS regioni utilizzando Global Datastore nella Guida per l'utente di ElastiCache](#).

- Per i API dettagli, consulta Command Reference. [DeleteGlobalReplicationGroup](#) AWS CLI

## delete-replication-group

Il seguente esempio di codice mostra come utilizzare `delete-replication-group`.

### AWS CLI

Per eliminare un gruppo di replica

L'`delete-replication-group` esempio seguente elimina un gruppo di replica esistente. Per impostazione predefinita, questa operazione elimina l'intero gruppo di replica, incluse le repliche primarie/primarie e tutte le repliche di lettura. Se il gruppo di replica ha un solo primario, è possibile facoltativamente eliminare solo le repliche di lettura, mantenendo la principale impostando `=true`. `RetainPrimaryCluster`

Quando ricevi una risposta corretta da questa operazione, Amazon inizia ElastiCache immediatamente a eliminare le risorse selezionate; non puoi annullare o annullare questa operazione. Valido solo per Redis.

```
aws elasticache delete-replication-group \  
--replication-group-id "mygroup"
```

Output:

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "mygroup",  
    "Description": "my group",  
    "Status": "deleting",  
    "PendingModifiedValues": {},  
    "AutomaticFailover": "disabled",  
    "SnapshotRetentionLimit": 0,  
    "SnapshotWindow": "06:00-07:00",  
    "TransitEncryptionEnabled": false,  
    "AtRestEncryptionEnabled": false  
  }  
}
```

- Per API i dettagli, vedere [DeleteReplicationGroup](#) in AWS CLI Command Reference.

## delete-snapshot

Il seguente esempio di codice mostra come utilizzare `delete-snapshot`.

### AWS CLI

Per eliminare uno snapshot

L'esempio seguente ha eliminato un'istantanea utilizzando il motore Redis.

```
aws elasticache delete-snapshot \  
  --snapshot-name mysnapshot
```

Output:

```
{  
  "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"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Backup and Restore ElastiCache for Redis](#) nella Guida per l'utente di ElastiCache.

- Per API i dettagli, consulta Command [DeleteSnapshotReference](#) AWS CLI .

## delete-user-group

Il seguente esempio di codice mostra come utilizzare `delete-user-group`.

### AWS CLI

Per eliminare un gruppo di utenti

L'esempio seguente elimina un gruppo di utenti.

```
aws elasticache delete-user-group \  
--user-group-id myusergroup
```

Output:

```
{  
  "UserGroupId": "myusergroup",  
  "Status": "deleting",  
  "Engine": "redis",  
  "UserIds": [  
    "default"  
  ],  
  "ReplicationGroups": [],  
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxxx52:usergroup:myusergroup"  
}
```

Per ulteriori informazioni, consulta [Authenticating Users with Role-Based Access Control \(RBAC\)](#) nella Guida per l'utente di Elasticache.

- Per i API dettagli, consulta Command Reference. [DeleteUserGroup](#) AWS CLI

## delete-user

Il seguente esempio di codice mostra come utilizzare `delete-user`.

### AWS CLI

Per eliminare un utente

L'`delete-user` esempio seguente elimina un utente.

```
aws elasticache delete-user \  
--user-id user2
```

Output:

```
{  
  "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"
}
```

Per ulteriori informazioni, consulta [Authenticating Users with Role-Based Access Control \(RBAC\)](#) nella Guida per l'utente di Elasticache.

- Per i API dettagli, consulta Command Reference. [DeleteUser](#) AWS CLI

## describe-cache-clusters

Il seguente esempio di codice mostra come utilizzare `describe-cache-clusters`.

### AWS CLI

Per descrivere un cluster di cache

L'esempio seguente descrive un cluster di cache.

```
aws elasticache describe-cache-clusters
```

Output:

```
{
  "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:xxxxxxxxxxx152: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:xxxxxxxxxxx152:cluster:my-cache-
cluster",
"ReplicationGroupLogDeliveryEnabled": false,
"LogDeliveryConfigurations": [
  {
    "LogType": "slow-log",
    "DestinationType": "cloudwatch-logs",
    "DestinationDetails": {
      "CloudWatchLogsDetails": {
        "LogGroup": "test-log"
      }
    },
    "LogFormat": "text",
    "Status": "active"
  }
]
}
```



```
}
```

Per ulteriori informazioni, consulta [Managing Clusters](#) nella Guida per l'utente di ElastiCache.

- Per API i dettagli, consulta Command [DescribeCacheClusters](#)Reference AWS CLI .

## describe-cache-engine-versions

Il seguente esempio di codice mostra come utilizzare `describe-cache-engine-versions`.

### AWS CLI

Per descrivere una versione del motore di cache

L'`describe-cache-engine-versions`esempio seguente restituisce un elenco dei motori di cache disponibili e delle relative versioni.

```
aws elasticache describe-cache-engine-versions \  
  --engine "Redis"
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

```
    "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"
}
]
```

```
}
```

- Per API i dettagli, vedere [DescribeCacheEngineVersions](#) in AWS CLI Command Reference.

## describe-cache-parameter-groups

Il seguente esempio di codice mostra come utilizzare `describe-cache-parameter-groups`.

### AWS CLI

Per descrivere un gruppo di parametri della cache

L'`describe-cache-parameter-groups` esempio seguente restituisce un elenco di descrizioni dei gruppi di parametri della cache.

```
aws elasticache describe-cache-parameter-groups \  
  --cache-parameter-group-name "mygroup"
```

Output:

```
{  
  "CacheParameterGroups": [  
    {  
      "CacheParameterGroupName": "mygroup",  
      "CacheParameterGroupFamily": "redis5.0",  
      "Description": " "  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Configurazione dei parametri del motore utilizzando i gruppi di parametri nella Guida](#) per l'utente di Elasticache.

- Per API i dettagli, vedere [DescribeCacheParameterGroups](#) in AWS CLI Command Reference.

## describe-cache-parameters

Il seguente esempio di codice mostra come utilizzare `describe-cache-parameters`.

### AWS CLI

Per descrivere i parametri della cache

Il seguente esempio "describe-cache-parameters" restituisce l'elenco dettagliato dei parametri per il gruppo di parametri di cache specificato.

```
aws elasticache describe-cache-parameters \  
  --cache-parameter-group-name "myparamgroup"
```

Output:

```
{  
  "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,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": "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Parameter Management nella Guida](#) per l'utente di ElastiCache.

- Per API i dettagli, consulta AWS CLI Command [DescribeCacheParameters](#) Reference.

## describe-cache-subnet-groups

Il seguente esempio di codice mostra come utilizzare `describe-cache-subnet-groups`.

AWS CLI

Per descrivere i gruppi di sottoreti della cache

L'`describe-cache-subnet-groups` seguente restituisce un elenco di gruppi di sottoreti.

```
aws elasticache describe-cache-subnet-groups
```

Output:

```

{
  "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"
        }
      }
    ]
  }
]
}

```

Per ulteriori informazioni, consulta [Subnet and Subnet Groups](#) nella Elasticache User Guide o [Subnet and Subnet Groups nella for Memcached](#) User Guide. ElastiCache

- Per AWS CLI i API dettagli, [DescribeCacheSubnetGroups](#) consulta Command Reference.

## describe-engine-default-parameters

Il seguente esempio di codice mostra come utilizzare `describe-engine-default-parameters`.

### AWS CLI

Per descrivere i parametri predefiniti del motore

L'`describe-engine-default-parameters` esempio seguente restituisce le informazioni predefinite sul motore e sui parametri di sistema per il motore di cache specificato.

```

aws elasticache describe-engine-default-parameters \
  --cache-parameter-group-family redis5.0

```

Output:

```

{
  "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-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,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": "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"
    }
]
}
}

```

- Per API i dettagli, vedere [DescribeEngineDefaultParameters](#) in AWS CLI Command Reference.

## describe-events

Il seguente esempio di codice mostra come utilizzare `describe-events`.

### AWS CLI

Per descrivere gli eventi di un gruppo di replica

L'`describe-events` esempio seguente restituisce un elenco di eventi per un gruppo di replica.

```

aws elasticache describe-events \
  --source-identifier test-cluster \
  --source-type replication-group

```

Output:

```

{
  "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"
    }
  ]
}

```

```
}
```

Per ulteriori informazioni, consulta [Monitoring Events nella Guida](#) per l'utente di Elasticache.

- Per API i dettagli, consulta AWS CLI Command [DescribeEvents](#)Reference.

## **describe-global-replication-groups**

Il seguente esempio di codice mostra come utilizzare `describe-global-replication-groups`.

### AWS CLI

Per descrivere i gruppi di replica globali

L'`describe-global-replication-groups`esempio seguente restituisce i dettagli di un datastore globale.

```
aws elasticache describe-global-replication-groups \  
  --global-replication-group-id my-grg
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Replica tra AWS regioni utilizzando Global Datastore nella Guida per l'utente di Elasticache](#).

- Per i API dettagli, consulta Command Reference. [DescribeGlobalReplicationGroups](#)AWS CLI

## describe-replication-groups

Il seguente esempio di codice mostra come utilizzare `describe-replication-groups`.

### AWS CLI

Per restituire un elenco di dettagli del gruppo di replica

L'`describe-replication-groups` seguente restituisce i gruppi di replica.

```
aws elasticache describe-replication-groups
```

Output:

```
{
  "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.xxxih.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.xxxih.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:xxxxxxxxxxx152:replicationgroup:my-cluster",
    "LogDeliveryConfigurations": [
      {
        "LogType": "slow-log",
        "DestinationType": "cloudwatch-logs",
        "DestinationDetails": {
          "CloudWatchLogsDetails": {
            "LogGroup": "test-log"
          }
        },
        "LogFormat": "json",
        "Status": "active"
      }
    ]
  }
]
}

```

Per ulteriori informazioni, consulta [Managing Clusters](#) nella Guida per l'utente di ElastiCache.

- Per API i dettagli, consulta Command [DescribeReplicationGroups](#) Reference AWS CLI .

## describe-reserved-cache-nodes-offerings

Il seguente esempio di codice mostra come utilizzare `describe-reserved-cache-nodes-offerings`.

### AWS CLI

Descrivere `reserved-cache-nodes-offerings`

L'`describe-reserved-cache-nodes-offerings` esempio seguente restituisce i dettagli di un' `reserved-cache-node` opzione.

**aws elasticache describe-reserved-cache-nodes-offerings**

Output:

```
{
  "ReservedCacheNodesOfferings": [
    {
      "ReservedCacheNodesOfferingId": "01ce0a19-a476-41cb-8aee-48eacbcd8e5",
      "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"
        }
      ]
    },
    ...
  ]
}
```

Per ulteriori informazioni, consulta [Ottenere informazioni sulle offerte di nodi riservati](#) nella Guida per l'utente di Elasticache Redis o [Informazioni sulle offerte di nodi riservati](#) nella Guida per l'utente di Elasticache Memcached.

- Per AWS CLI i API dettagli, [DescribeReservedCacheNodesOfferings](#) consulta Command Reference.

## describe-reserved-cache-nodes

Il seguente esempio di codice mostra come utilizzare `describe-reserved-cache-nodes`.

### AWS CLI

Per descrivere i nodi di cache riservati

L'`describe-reserved-cache-nodes` esempio seguente restituisce informazioni sui nodi di cache riservati per questo account o sul nodo di cache riservato specificato.

```
aws elasticache describe-reserved-cache-nodes
```

Output:

```
{
  "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:xxxxxxxxxxx52:reserved-instance:mynode"
    }
]
}

```

Per ulteriori informazioni, consulta [Managing Costs with Reserved Nodes](#) nella Guida per l'utente di Elasticache.

- Per API i dettagli, consulta AWS CLI Command [DescribeReservedCacheNodes](#) Reference.

## describe-service-updates

Il seguente esempio di codice mostra come utilizzare `describe-service-updates`.

### AWS CLI

Per descrivere gli aggiornamenti del servizio

L'`describe-service-updates` esempio seguente restituisce i dettagli sugli aggiornamenti del servizio.

```
aws elasticache describe-service-updates
```

Output:

```

{
  "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"
    }
  ]
}

```

- Per API i dettagli, vedere [DescribeServiceUpdates](#) in AWS CLI Command Reference.

## describe-snapshots

Il seguente esempio di codice mostra come utilizzare `describe-snapshots`.

### AWS CLI

Per descrivere le istantanee

Il seguente esempio «`describe-snapshots`» restituisce informazioni sulle istantanee del cluster o del gruppo di replica.

```
aws elasticache describe-snapshots
```

Output:

```

{
  "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:xxxxxxxxxx52: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"
      }
    ]
  }
]
```



Per ulteriori informazioni, consulta [Backup and Restore ElastiCache for Redis](#) nella Guida per l'utente di Elasticache.

- Per API i dettagli, consulta Command [DescribeSnapshots](#) Reference AWS CLI .

## describe-update-actions

Il seguente esempio di codice mostra come utilizzare `describe-update-actions`.

### AWS CLI

Per descrivere le azioni di aggiornamento

L'`describe-update-actions` esempio seguente restituisce i dettagli delle azioni di aggiornamento.

```
aws elasticache describe-update-actions
```

Output:

```
{
  "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"
}
]
}
```

Per ulteriori informazioni, consulta [Self-Service Updates in Amazon ElastiCache nella Elasticache User Guide](#).

- Per API i dettagli, consulta Command [DescribeUpdateActions](#)Reference AWS CLI .

## describe-user-groups

Il seguente esempio di codice mostra come utilizzare `describe-user-groups`.

### AWS CLI

Per descrivere i gruppi di utenti

L'`describe-user-groups`esempio seguente restituisce un elenco di gruppi di utenti.

```
aws elasticache describe-user-groups
```

Output:

```
{
  "UserGroups": [
    {
      "UserGroupId": "myusergroup",
      "Status": "active",
      "Engine": "redis",
      "UserIds": [
        "default"
      ],
      "ReplicationGroups": [],
      "ARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxx52:usergroup:myusergroup"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Authenticating Users with Role-Based Access Control \(RBAC\)](#) nella Guida per l'utente di Elasticache.

- Per i API dettagli, consulta Command Reference. [DescribeUserGroups](#)AWS CLI

## describe-users

Il seguente esempio di codice mostra come utilizzare `describe-users`.

### AWS CLI

Per descrivere gli utenti

L'`describe-users`esempio seguente restituisce un elenco di utenti.

```
aws elasticache describe-users
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Authenticating Users with Role-Based Access Control \(RBAC\)](#) nella Guida per l'utente di Elasticache.

- Per i API dettagli, consulta Command Reference. [DescribeUsers](#) AWS CLI

## disassociate-global-replication-group

Il seguente esempio di codice mostra come utilizzare `disassociate-global-replication-group`.

AWS CLI

Per disassociare un cluster secondario da un gruppo di replica globale

L'esempio seguente rimuove un cluster secondario da un datastore globale

```

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

```

Output:

```
{
```

```

"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
}
}

```

Per ulteriori informazioni, consulta [Replica tra AWS regioni utilizzando Global Datastore nella Guida per l'utente di ElastiCache](#).

- Per i API dettagli, consulta Command Reference. [DisassociateGlobalReplicationGroup](#) AWS CLI

## increase-node-groups-in-global-replication-group

Il seguente esempio di codice mostra come utilizzare `increase-node-groups-in-global-replication-group`.

AWS CLI

Per aumentare il numero di gruppi di nodi in un gruppo di replica globale

Quanto segue `increase-node-groups-in-global-replication-group` aumenta il numero dei gruppi di nodi utilizzando il motore Redis.

```
aws elasticache increase-node-groups-in-global-replication-group \  
  --global-replication-group-id sgai-pat-test-4 \  
  --node-group-count 6 \  
  --apply-immediately
```

Output:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

```

    },
    {
      "GlobalNodeGroupId": "sgaui-test-4-0003",
      "Slots": "10923-11190,13375-16383"
    },
    {
      "GlobalNodeGroupId": "sgaui-test-4-0004",
      "Slots": "235-2419,5905-6996"
    },
    {
      "GlobalNodeGroupId": "sgaui-test-4-0005",
      "Slots": "9831-10922,11191-13374"
    }
  ],
  "AuthTokenEnabled": false,
  "TransitEncryptionEnabled": false,
  "AtRestEncryptionEnabled": false
}
}

```

Per ulteriori informazioni, consulta [Replica tra AWS regioni utilizzando Global Datastore nella Guida per l'utente di ElastiCache](#).

- Per i API dettagli, consulta Command Reference.  
[IncreaseNodeGroupsInGlobalReplicationGroup](#) AWS CLI

## increase-replica-count

Il seguente esempio di codice mostra come utilizzare `increase-replica-count`.

### AWS CLI

Per aumentare il numero di repliche

L'`increase-replica-count` esempio seguente esegue una delle due operazioni. Può aumentare dinamicamente il numero di repliche in un gruppo di replica Redis (modalità cluster disabilitata). Oppure può aumentare dinamicamente il numero di nodi di replica in uno o più gruppi di nodi (shard) di un gruppo di replica Redis (abilitata in modalità cluster). Questa operazione viene eseguita senza tempi di inattività del cluster.

```

aws elasticache increase-replica-count \
  --replication-group-id "my-cluster" \

```





```

        {
            "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
}
}

```

Per ulteriori informazioni, consulta [Aumentare il numero di repliche in uno shard nella Guida](#) per l'utente di Elasticache.

- Per i API dettagli, consulta [IncreaseReplicaCount](#) Command Reference.AWS CLI

## list-allowed-node-type-modifications

Il seguente esempio di codice mostra come utilizzare `list-allowed-node-type-modifications`.

### AWS CLI

Per elencare le modifiche consentite ai nodi

L'`list-allowed-node-type-modifications`esempio seguente elenca tutti i tipi di nodi disponibili su cui è possibile scalare il tipo di nodo corrente del cluster Redis o del gruppo di replica.

```
aws elasticache list-allowed-node-type-modifications \
```

```
--replication-group-id "my-replication-group"
```

Output:

```
{
  "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"
  ]
}
```

Per ulteriori informazioni, consulta [Scaling ElastiCache for Redis Clusters nella Guida per l'utente di ElastiCache](#).

- Per i API dettagli, vedere in Command Reference. [ListAllowedNodeTypeModifications](#) AWS CLI

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una risorsa

L'`list-tags-for-resource` esempio seguente elenca i tag per una risorsa.

```
aws elasticache list-tags-for-resource \  
  --resource-name "arn:aws:elasticache:us-east-1:123456789012:cluster:my-cluster"
```

Output:

```
{  
  "TagList": [  
    {  
      "Key": "Project",  
      "Value": "querySpeedUp"  
    },  
    {  
      "Key": "Environment",  
      "Value": "PROD"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Listing Tags Using the AWS CLI nella Guida](#) per l'utente di Elasticache.

- Per API i dettagli, consulta AWS CLI Command [ListTagsForResource](#) Reference.

## modify-cache-cluster

Il seguente esempio di codice mostra come utilizzare `modify-cache-cluster`.

### AWS CLI

Per modificare i cluster di cache

L'`modify-cache-cluster` esempio seguente modifica le impostazioni per il cluster specificato.

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id "my-cluster" \  
  --num-cache-nodes 1
```

Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, consulta [Modificare un ElastiCache cluster](#) nella Guida per l'utente di ElastiCache.

- Per API i dettagli, consulta Command [ModifyCacheCluster](#) Reference AWS CLI .

## modify-cache-parameter-group

Il seguente esempio di codice mostra come utilizzare `modify-cache-parameter-group`.

### AWS CLI

Per modificare un gruppo di parametri della cache

L'`modify-cache-parameter-group`esempio seguente modifica i parametri del gruppo di parametri di cache specificato.

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name "mygroup" \  
  --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"
```

Output:

```
{  
  "CacheParameterGroupName": "mygroup"  
}
```

Per ulteriori informazioni, consulta [Modifica di un gruppo di parametri](#) nella Guida per l'utente di Elasticache.

- Per API i dettagli, consulta Command [ModifyCacheParameterGroup](#)Reference AWS CLI .

## modify-cache-subnet-group

Il seguente esempio di codice mostra come utilizzare `modify-cache-subnet-group`.

### AWS CLI

Per modificare un gruppo di sottoreti della cache

L'`modify-cache-subnet-group`esempio seguente modifica il gruppo di sottoreti di cache specificato.

```
aws elasticache modify-cache-subnet-group \  
  --cache-subnet-group-name kxkxk \  
  --cache-subnet-group-description "mygroup"
```

Output:

```
{
  "CacheSubnetGroup": {
    "CacheSubnetGroupName": "kxkxk",
    "CacheSubnetGroupDescription": "mygroup",
    "VpcId": "vpc-xxxxcdb",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-xxxxbff",
        "SubnetAvailabilityZone": {
          "Name": "us-west-2a"
        }
      }
    ]
  }
}
```

Per ulteriori informazioni, consulta [Modifica di un gruppo di sottoreti](#) nella Guida per l'utente di Elasticache.

- Per i API dettagli, consulta [ModifyCacheSubnetGroup](#) Command Reference.AWS CLI

## modify-global-replication-group

Il seguente esempio di codice mostra come utilizzare `modify-global-replication-group`.

### AWS CLI

Per modificare un gruppo di replica globale

Quanto segue `modify-global-replication-group` modifica le proprietà di un gruppo di replica globale, in questo caso disabilitando il failover automatico, utilizzando il motore Redis.

```
aws elasticache modify-global-replication-group \
  --global-replication-group-id sgai-pat-group \
  --apply-immediately \
  --no-automatic-failover-enabled
```

### Output

```
{
  "GlobalReplicationGroup": {
    "GlobalReplicationGroupId": "sgai-test-group",
```

```

    "GlobalReplicationGroupDescription": " ",
    "Status": "modifying",
    "CacheNodeType": "cache.r5.large",
    "Engine": "redis",
    "EngineVersion": "5.0.6",
    "ClusterEnabled": false,
    "AuthTokenEnabled": false,
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}

```

Per ulteriori informazioni, consulta [Replica tra AWS regioni utilizzando Global Datastore](#) nella Guida per l'utente di ElastiCache.

- Per i API dettagli, consulta Command Reference. [ModifyGlobalReplicationGroup](#) AWS CLI

## modify-replication-group-shard-configuration

Il seguente esempio di codice mostra come utilizzare `modify-replication-group-shard-configuration`.

### AWS CLI

Per modificare la configurazione di uno shard di un gruppo di replica

quanto segue `modify-replication-group-shard-configuration` riduce il numero dei gruppi di nodi utilizzando il motore Redis.

```

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
}
}
```

Per ulteriori informazioni, consulta [Scaling ElastiCache for Redis Clusters nella Guida per l'utente di ElastiCache](#).

- Per i API dettagli, vedere in Command Reference.

[ModifyReplicationGroupShardConfiguration](#) AWS CLI

## modify-replication-group

Il seguente esempio di codice mostra come utilizzare `modify-replication-group`.

### AWS CLI

Per modificare un gruppo di replica

Quanto segue `modify-replication-group` disabilita Multi-AZ utilizzando il motore Redis.

```
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
  }
}

```

Per ulteriori informazioni, consulta [Modifica di un gruppo di replica](#) nella Guida per l'utente di Elasticache.

- Per i API dettagli, consulta [ModifyReplicationGroup](#) Command Reference.AWS CLI

## modify-user-group

Il seguente esempio di codice mostra come utilizzare `modify-user-group`.

### AWS CLI

Per modificare un gruppo di utenti

L'`modify-user-group`esempio seguente aggiunge un utente al gruppo di utenti.

```

aws elasticache modify-user-group \
  --user-group-id myusergroup \
  --user-ids-to-add user1

```

Output:

```

{
  "UserGroupId": "myusergroup",
  "Status": "modifying",
  "Engine": "redis",
  "UserIds": [
    "default"
  ],
  "PendingChanges": {
    "UserIdsToAdd": [
      "user1"
    ]
  },
  "ReplicationGroups": [],
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"
}

```

```
}
```

Per ulteriori informazioni, consulta [Authenticating Users with Role-Based Access Control \(RBAC\)](#) nella Guida per l'utente di ElastiCache.

- Per i API dettagli, consulta Command Reference. [ModifyUserGroup](#) AWS CLI

## modify-user

Il seguente esempio di codice mostra come utilizzare `modify-user`.

### AWS CLI

Per modificare un utente

L'`modify-user` esempio seguente modifica la stringa di accesso di un utente.

```
aws elasticache modify-user \  
  --user-id user2 \  
  --append-access-string "on ~* +@all"
```

Output:

```
{  
  "UserId": "user2",  
  "UserName": "myUser",  
  "Status": "modifying",  
  "Engine": "redis",  
  "AccessString": "on ~* +@all",  
  "UserGroupIds": [],  
  "Authentication": {  
    "Type": "password",  
    "PasswordCount": 1  
  },  
  "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxxx52:user:user2"  
}
```

Per ulteriori informazioni, consulta [Authenticating Users with Role-Based Access Control \(RBAC\)](#) nella Guida per l'utente di ElastiCache.

- Per i API dettagli, consulta Command Reference. [ModifyUser](#) AWS CLI

## purchase-reserved-cache-nodes-offering

Il seguente esempio di codice mostra come utilizzare `purchase-reserved-cache-nodes-offering`.

### AWS CLI

Per acquistare un `reserved-cache-node-offering`

L'`purchase-reserved-cache-nodes-offering` seguente consente di acquistare un'offerta di nodi di cache riservati.

```
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": "xxxxxxx-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"  
      }  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Ottenere informazioni sulle offerte di nodi riservati](#) nella Guida per l'utente di Elasticache Redis o [Informazioni sulle offerte di nodi riservati](#) nella Guida per l'utente di Elasticache Memcached.



- Per AWS CLI i API dettagli, [PurchaseReservedCacheNodesOffering](#) consulta Command Reference.

## reboot-cache-cluster

Il seguente esempio di codice mostra come utilizzare `reboot-cache-cluster`.

### AWS CLI

Per riavviare un cluster di cache

L'`reboot-cache-cluster` esempio seguente riavvia alcuni o tutti i nodi di cache all'interno di un cluster di cui è stato eseguito il provisioning. Questa operazione applica tutti i gruppi di parametri di cache modificati al cluster. L'operazione di riavvio avviene il prima possibile e provoca un'interruzione momentanea del cluster. Durante il riavvio, lo stato del cluster è impostato su `REBOOTING`.

```
aws elasticache reboot-cache-cluster \  
  --cache-cluster-id "my-cluster-001" \  
  --cache-node-ids-to-reboot "0001"
```

Output:

```
{  
  "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
  }
}

```

Per ulteriori informazioni, consulta [Rebooting a Cluster < https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters\\_rebooting.html](https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters_rebooting.html) nella Guida per l'utente di Elasticache.

- Per i API dettagli, [RebootCacheCluster](#) consulta AWS CLI Command Reference.

## reset-cache-parameter-group

Il seguente esempio di codice mostra come utilizzare `reset-cache-parameter-group`.

### AWS CLI

Per reimpostare un gruppo di parametri della cache

L'esempio seguente modifica i parametri di un gruppo di parametri della cache in base al valore predefinito del motore o del sistema. È possibile ripristinare parametri specifici inviando un elenco di nomi di parametri. Per reimpostare l'intero gruppo di parametri della cache, specificate i `--cache-parameter-group-name parametri --reset-all-parameters` and.

```
aws elasticache reset-cache-parameter-group \  
  --cache-parameter-group-name "mygroup" \  
  --reset-all-parameters
```

Output:

```
{  
  "CacheParameterGroupName": "mygroup"  
}
```

- Per API i dettagli, consultate [ResetCacheParameterGroup AWS CLI Command Reference](#).

## start-migration

Il seguente esempio di codice mostra come utilizzare `start-migration`.

### AWS CLI

Per avviare una migrazione

Quanto segue `start-migration` migra i tuoi dati da Redis self-hosted su Amazon ad EC2 Amazon ElastiCache, utilizzando il motore Redis.

```
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
}
}

```

Per ulteriori informazioni, consulta la sezione [Migrazione online a ElastiCache nella Guida per l'utente di Elasticache](#).

- Per API i dettagli, consulta AWS CLI Command [StartMigration](#) Reference.

## test-failover

Il seguente esempio di codice mostra come utilizzare `test-failover`.

### AWS CLI

Per testare il failover di un gruppo di nodi

L'`test-failover` seguente verifica il failover automatico sul gruppo di nodi specificato (chiamato shard nella console) in un gruppo di replica (chiamato cluster nella console).

```

aws elasticache test-failover /
  --replication-group-id "mycluster" /
  --node-group-id "0001"

```

Output:

```

{
  "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
  }
}
```

- Per API i dettagli, vedere [TestFailover](#) in AWS CLI Command Reference.

## MediaStore esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with MediaStore.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-container**

Il seguente esempio di codice mostra come utilizzare `create-container`.

AWS CLI

Per creare un contenitore

L'`create-container` esempio seguente crea un nuovo contenitore vuoto.

```
aws mediastore create-container --container-name ExampleContainer
```

Output:



```
{
  "Container": {
    "AccessLoggingEnabled": false,
    "CreationTime": 1563557265,
    "Name": "ExampleContainer",
    "Status": "CREATING",
    "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer"
  }
}
```

Per ulteriori informazioni, consulta [Creazione di un contenitore](#) nella Guida MediaStore utente di AWS Elemental.

- Per API i dettagli, consultate [CreateContainer AWS CLI Command Reference](#).

## delete-container-policy

Il seguente esempio di codice mostra come utilizzare `delete-container-policy`.

### AWS CLI

Per eliminare una politica relativa ai contenitori

L'esempio seguente elimina la politica assegnata al contenitore specificato. Quando la politica viene eliminata, AWS Elemental assegna MediaStore automaticamente la politica predefinita al contenitore.

```
aws mediastore delete-container-policy \
  --container-name LiveEvents
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la guida di riferimento [DeleteContainerPolicy](#) di AWS MediaStore API Elemental.

- Per API i dettagli, vedere [DeleteContainerPolicy](#) in AWS CLI Command Reference.

## delete-container

Il seguente esempio di codice mostra come utilizzare `delete-container`.

## AWS CLI

Per eliminare un contenitore

L'`delete-container` esempio seguente elimina il contenitore specificato. Puoi eliminare un container solo se non contiene oggetti.

```
aws mediastore delete-container \  
  --container-name=ExampleLiveDemo
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di un contenitore](#) nella Guida per l'utente di AWS MediaStore Elemental.

- Per API i dettagli, consultate AWS CLI Command [DeleteContainer](#) Reference.

## `delete-cors-policy`

Il seguente esempio di codice mostra come utilizzare `delete-cors-policy`.

### AWS CLI

Per eliminare una CORS politica

L'`delete-cors-policy` esempio seguente elimina la politica di condivisione delle risorse tra le origini (CORS) assegnata al contenitore specificato.

```
aws mediastore delete-cors-policy \  
  --container-name ExampleContainer
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Eliminazione di una CORS politica](#) nella Guida per l'utente di AWS MediaStore Elemental.

- Per API i dettagli, consultate AWS CLI Command [DeleteCorsPolicy](#) Reference.

## `delete-lifecycle-policy`

Il seguente esempio di codice mostra come utilizzare `delete-lifecycle-policy`.

## AWS CLI

Per eliminare una politica del ciclo di vita degli oggetti

L'`delete-lifecycle-policy` esempio seguente elimina la politica del ciclo di vita degli oggetti allegata al contenitore specificato. Questa modifica può richiedere fino a 20 minuti per avere effetto.

```
aws mediastore delete-lifecycle-policy \  
  --container-name LiveEvents
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di una policy sul ciclo di vita degli oggetti](#) nella Guida per l'utente di AWS MediaStore Elemental.

- Per i API dettagli, vedere [DeleteLifecyclePolicy](#) in Command Reference.AWS CLI

## `describe-container`

Il seguente esempio di codice mostra come utilizzare `describe-container`.

## AWS CLI

Per visualizzare i dettagli di un contenitore

L'`describe-container` esempio seguente visualizza i dettagli del contenitore specificato.

```
aws mediastore describe-container \  
  --container-name ExampleContainer
```

Output:

```
{  
  "Container": {  
    "CreationTime": 1563558086,  
    "AccessLoggingEnabled": false,  
    "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/  
ExampleContainer",  
    "Status": "ACTIVE",  
    "Name": "ExampleContainer",
```



## AWS CLI

Per visualizzare una politica del contenitore

L'`get-container-policy` seguente mostra la politica basata sulle risorse del contenitore specificato.

```
aws mediastore get-container-policy \  
  --container-name ExampleLiveDemo
```

Output:

```
{  
  "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"  
          }  
        }  
      }  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Visualizzazione di una politica sui contenitori](#) nella Guida per l'utente di AWS MediaStore Elemental.

- Per API i dettagli, consultate [GetContainerPolicy AWS CLI Command Reference](#).

## get-cors-policy

Il seguente esempio di codice mostra come utilizzare `get-cors-policy`.

### AWS CLI

Per visualizzare una CORS politica

L'esempio seguente visualizza la politica di condivisione delle risorse tra le origini (CORS) assegnata al contenitore specificato.

```
aws mediastore get-cors-policy \  
  --container-name ExampleContainer \  
  --region us-west-2
```

Output:

```
{  
  "CorsPolicy": [  
    {  
      "AllowedMethods": [  
        "GET",  
        "HEAD"  
      ],  
      "MaxAgeSeconds": 3000,  
      "AllowedOrigins": [  
        ""  
      ],  
      "AllowedHeaders": [  
        ""  
      ]  
    }  
  ]  
}
```

Per ulteriori informazioni, vedete [Visualizzazione di una CORS politica nella Guida](#) per l'utente di AWS Elemental MediaStore.

- Per API i dettagli, consultate [GetCorsPolicy AWS CLI Command Reference](#).

## get-lifecycle-policy

Il seguente esempio di codice mostra come utilizzare `get-lifecycle-policy`.

## AWS CLI

Per visualizzare una politica del ciclo di vita degli oggetti

L'get-lifecycle-policy esempio seguente visualizza la politica del ciclo di vita degli oggetti allegata al contenitore specificato.

```
aws mediastore get-lifecycle-policy \  
  --container-name LiveEvents
```

Output:

```
{  
  "LifecyclePolicy": {  
    "rules": [  
      {  
        "definition": {  
          "path": [  
            {  
              "prefix": "Football/"  
            },  
            {  
              "prefix": "Baseball/"  
            }  
          ],  
          "days_since_create": [  
            {  
              "numeric": [  
                ">",  
                28  
              ]  
            }  
          ],  
          "action": "EXPIRE"  
        }  
      ]  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Visualizzazione di una politica del ciclo di vita degli oggetti nella Guida per l'utente di ElementalAWS . MediaStore](#)

- Per API i dettagli, vedere [GetLifecyclePolicy](#) in AWS CLI Command Reference.

## get-object

Il seguente esempio di codice mostra come utilizzare `get-object`.

### AWS CLI

Per scaricare un oggetto

L'`get-object` esempio seguente scarica un oggetto nell'endpoint specificato.

```
aws mediastore-data get-object \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path=/folder_name/README.md README.md
```

Output:

```
{  
  "ContentLength": "2307346",  
  "ContentType": "image/jpeg",  
  "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",  
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f5555555555555555da6d3",  
  "StatusCode": 200  
}
```

Per scaricare parte di un oggetto

L'`get-object` esempio seguente scarica una parte di un oggetto sull'endpoint specificato.

```
aws mediastore-data get-object \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path /folder_name/README.md \  
  --range="bytes=0-100" README2.md
```

Output:

```
{  
  "StatusCode": 206,  
  "ContentRange": "bytes 0-100/2307346",  
  "ContentLength": "101",
```



```
"LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
"ContentType": "image/jpeg",
"ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f5555555555555555da6d3"
}
```

Per ulteriori informazioni, consultate [Downloading an Object nella Guida](#) per l' MediaStore utente di AWS Elemental.

- Per API i dettagli, consultate [GetObject AWS CLI Command Reference](#).

## list-containers

Il seguente esempio di codice mostra come utilizzare `list-containers`.

### AWS CLI

Per visualizzare un elenco di contenitori

L'`list-containers` esempio seguente visualizza un elenco di tutti i contenitori associati all'account.

```
aws mediastore list-containers
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione di un elenco di contenitori](#) nella Guida per l' MediaStore utente di AWS Elemental.

- Per API i dettagli, consultate [ListContainers AWS CLI Command Reference](#).

## list-items

Il seguente esempio di codice mostra come utilizzare `list-items`.

### AWS CLI

Esempio 1: per visualizzare un elenco di oggetti e cartelle in un contenitore specifico

L'`list-items`esempio seguente visualizza gli elementi (oggetti e cartelle) memorizzati nel contenitore specificato.

```

aws mediastore-data list-items \
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com

```

Output:

```

{
  "Items": [
    {
      "ContentType": "image/jpeg",
      "LastModified": 1563571859.379,
      "Name": "filename.jpg",
      "Type": "OBJECT",
      "ETag":
"543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
      "ContentLength": 3784
    },
    {
      "Type": "FOLDER",

```

```

        "Name": "ExampleLiveDemo"
      }
    ]
  }

```

Esempio 2: Per visualizzare un elenco di oggetti e cartelle in una cartella specifica

L'`list-items`esempio seguente visualizza gli elementi (oggetti e cartelle) memorizzati in una cartella specifica.

```

aws mediastore-data list-items \
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com

```

Output:

```

{
  "Items": [
    {
      "ContentType": "image/jpeg",
      "LastModified": 1563571859.379,
      "Name": "filename.jpg",
      "Type": "OBJECT",
      "ETag":
"543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
      "ContentLength": 3784
    },
    {
      "Type": "FOLDER",
      "Name": "ExampleLiveDemo"
    }
  ]
}

```

Per ulteriori informazioni, vedete [Visualizzazione di un elenco di oggetti](#) nella Guida per l' MediaStore utente di AWS Elemental.

- Per API i dettagli, consultate [ListItems AWS CLI](#) Command Reference.

## **list-tags-for-resource**

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

## AWS CLI

Per elencare i tag per un contenitore

L'`list-tags-for-resource` seguente visualizza le chiavi e i valori dei tag assegnati al contenitore specificato.

```
aws mediastore list-tags-for-resource \
  --resource arn:aws:mediastore:us-west-2:1213456789012:container/ExampleContainer
```

Output:

```
{
  "Tags": [
    {
      "Value": "Test",
      "Key": "Environment"
    },
    {
      "Value": "West",
      "Key": "Region"
    }
  ]
}
```

Per ulteriori informazioni, [ListTagsForResource](#) consultate la sezione [MediaStore API Riferimento AWS elementare](#).

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## **put-container-policy**

Il seguente esempio di codice mostra come utilizzare `put-container-policy`.

## AWS CLI

Per modificare una politica del contenitore

L'`put-container-policy` seguente assegna una politica diversa al contenitore specificato. In questo esempio, la politica aggiornata viene definita in un file denominato `LiveEventsContainerPolicy.json`.

```
aws mediastore put-container-policy \  
  --container-name LiveEvents \  
  --policy file://LiveEventsContainerPolicy.json
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Modifica di una policy sui contenitori](#) nella Guida per l' MediaStore utente di AWS Elemental.

- Per API i dettagli, consultate [PutContainerPolicy AWS CLI Command Reference](#).

## put-cors-policy

Il seguente esempio di codice mostra come utilizzare `put-cors-policy`.

### AWS CLI

Esempio 1: aggiungere una CORS politica

L'`put-cors-policy` seguente aggiunge una politica di condivisione delle risorse (CORS) tra origini diverse al contenitore specificato. Il contenuto della CORS policy si trova nel file denominato `corsPolicy.json`.

```
aws mediastore put-cors-policy \  
  --container-name ExampleContainer \  
  --cors-policy file://corsPolicy.json
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere una CORS politica a un contenitore](#) nella Guida per l' MediaStore utente di AWS Elemental.

Esempio 2: Per modificare una politica CORS

L'`put-cors-policy` seguente aggiorna la politica di condivisione delle risorse tra le origini (CORS) assegnata al contenitore specificato. Il contenuto della CORS politica aggiornata si trova nel file denominato `corsPolicy2.json`.

Per ulteriori informazioni, consulta [Modifica di una CORS politica](#) nella Guida per l' MediaStore utente di AWS Elemental.

- Per API i dettagli, consultate [PutCorsPolicy AWS CLI Command Reference](#).

## put-lifecycle-policy

Il seguente esempio di codice mostra come utilizzare `put-lifecycle-policy`.

### AWS CLI

Per creare una politica del ciclo di vita degli oggetti

L'`put-lifecycle-policy` esempio seguente allega una politica del ciclo di vita degli oggetti al contenitore specificato. Ciò consente di specificare per quanto tempo il servizio deve archiviare gli oggetti nel contenitore. MediaStore elimina gli oggetti nel contenitore una volta raggiunta la data di scadenza, come indicato nella policy, che si trova nel file denominato `LiveEventsLifecyclePolicy.json`.

```
aws mediastore put-lifecycle-policy \  
  --container-name ExampleContainer \  
  --lifecycle-policy file://ExampleLifecyclePolicy.json
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere una policy sul ciclo di vita degli oggetti a un contenitore nella Guida per l'utente di AWS MediaStore Elemental](#).

- Per API i dettagli, consulta Command [PutLifecyclePolicy](#) Reference AWS CLI .

## put-object

Il seguente esempio di codice mostra come utilizzare `put-object`.

### AWS CLI

Per caricare un oggetto

L'`put-object` esempio seguente carica un oggetto nel contenitore specificato. È possibile specificare il percorso della cartella in cui salvare l'oggetto all'interno del contenitore. Se la cartella esiste già, AWS Elemental MediaStore memorizza l'oggetto nella cartella. Se la cartella non esiste, il servizio la crea e quindi memorizza l'oggetto nella cartella.

```
aws mediastore-data put-object \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --body README.md \  
  --path /folder_name/README.md \  
  --signature signature
```

```
--cache-control "max-age=6, public" \  
--content-type binary/octet-stream
```

Output:

```
{  
  "ContentSHA256":  
    "74b5fdb517f423ed750ef214c44adfe2be36e37d861eafe9c842cbe1bf387a9d",  
  "StorageClass": "TEMPORAL",  
  "ETag": "af3e4731af032167a106015d1f2fe934e68b32ed1aa297a9e325f5c64979277b"  
}
```

Per ulteriori informazioni, consulta [Caricamento di un oggetto nella Guida](#) per l' MediaStore utente di AWS Elemental.

- Per API i dettagli, consultate AWS CLI Command [PutObjectReference](#).

## start-access-logging

Il seguente esempio di codice mostra come utilizzare `start-access-logging`.

AWS CLI

Per abilitare la registrazione degli accessi su un contenitore

L'`start-access-logging` seguente abilita la registrazione degli accessi sul contenitore specificato.

```
aws mediastore start-access-logging \  
--container-name LiveEvents
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Enabling Access Logging for a Container nella Guida](#) per l'utente di AWS MediaStore Elemental.

- Per API i dettagli, consultate AWS CLI Command [StartAccessLoggingReference](#).

## stop-access-logging

Il seguente esempio di codice mostra come utilizzare `stop-access-logging`.

## AWS CLI

Per disabilitare la registrazione degli accessi su un contenitore

L'`stop-access-logging`esempio seguente disabilita la registrazione degli accessi sul contenitore specificato.

```
aws mediastore stop-access-logging \  
  --container-name LiveEvents
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Disabling Access Logging for a Container nella ElementarAWS User Guide](#). MediaStore

- Per API i dettagli, consultate Command [StopAccessLoggingReference AWS CLI](#) .

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

## AWS CLI

Per aggiungere tag a un contenitore

L'`tag-resource`esempio seguente aggiunge chiavi e valori di tag al contenitore specificato.

```
aws mediastore tag-resource \  
  --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer \  
  --tags '[{"Key": "Region", "Value": "West"}, {"Key": "Environment", "Value":  
  "Test"}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [TagResource](#)consultate la sezione [MediaStore API Riferimento AWS elementare](#).

- Per API i dettagli, vedere [TagResource](#)in [AWS CLI Command Reference](#).

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.



## AWS CLI

Per rimuovere i tag da un contenitore

L'untag-resourceesempio seguente rimuove la chiave di tag specificata e il valore associato da un contenitore.

```
aws mediastore untag-resource \  
  --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer \  
  --tag-keys Region
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [UntagResource](#)consultate la sezione [MediaStore API](#)Riferimento AWS elementare. .

- Per API i dettagli, vedere [UntagResource](#)in AWS CLI Command Reference.

## EMREsempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con AmazonEMR.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **add-instance-fleet**

Il seguente esempio di codice mostra come utilizzareadd-instance-fleet.

## AWS CLI

Per aggiungere un parco di istanze di attività a un cluster

Questo esempio aggiunge un nuovo parco di istanze di attività al cluster specificato.

Comando:

```
aws emr add-instance-fleet --cluster-id 'j-12ABCDEFGHI34JK' --instance-fleet
InstanceFleetType=TASK,TargetSpotCapacity=1,LaunchSpecifications={SpotSpecification='{Timeo
```

Output:

```
{
  "ClusterId": "j-12ABCDEFGHI34JK",
  "InstanceFleetId": "if-23ABCDEFGHI45JJ"
}
```

- Per API i dettagli, vedere [AddInstanceFleet](#) in AWS CLI Command Reference.

## add-steps

Il seguente esempio di codice mostra come utilizzare `add-steps`.

## AWS CLI

1. Per aggiungere JAR passaggi personalizzati a un cluster

Comando:

```
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
```

Parametri richiesti:

```
Jar
```

Parametri opzionali:

```
Type, Name, ActionOnFailure, Args
```

### Output:

```
{
  "StepIds": [
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

## 2. Per aggiungere passaggi di streaming a un cluster

### Comando:

```
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]
```

### Parametri richiesti:

```
Type, Args
```

### Parametri opzionali:

```
Name, ActionOnFailure
```

### JSONequivalente (contenuto di 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 gli argomenti devono includere opzioni e valori come elementi propri nell'elenco.

Comando (usando step.json):

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps file:///./step.json
```

Output:

```
{
  "StepIds": [
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

3. Per aggiungere una fase di streaming con più file a un cluster (JSONsolo)

JSON(multiplefiles.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"
    ]
  }
]
```

Comando:

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps file:///./multiplefiles.json
```

**Parametri richiesti:**

```
Type, Args
```

**Parametri opzionali:**

```
Name, ActionOnFailure
```

**Output:**

```
{
  "StepIds": [
    "s-XXXXXXXX",
  ]
}
```

**4. Per aggiungere passaggi Hive a un cluster****Comando:**

```
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]
```

**Parametri richiesti:**

```
Type, Args
```

**Parametri opzionali:**

```
Name, ActionOnFailure
```

**Output:**

```
{
  "StepIds": [
    "s-XXXXXXXX",
  ]
}
```

```

    "s-YYYYYYYY"
  ]
}

```

## 5. Per aggiungere Pig steps a un cluster

Comando:

```

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]

```

Parametri richiesti:

Type, Args

Parametri opzionali:

Name, ActionOnFailure

Output:

```

{
  "StepIds":[
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}

```

## 6. Per aggiungere i passaggi di Impala a un cluster

Comando:

```

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

```

Parametri richiesti:

```
Type, Args
```

Parametri opzionali:

```
Name, ActionOnFailure
```

Output:

```
{
  "StepIds": [
    "s-XXXXXXXX",
    "s-YYYYYYYY"
  ]
}
```

- Per API i dettagli, vedere [AddSteps](#) in AWS CLI Command Reference.

## add-tags

Il seguente esempio di codice mostra come utilizzare `add-tags`.

AWS CLI

1. Per aggiungere tag a un cluster

Comando:

```
aws emr add-tags --resource-id j-xxxxxxx --tags name="John Doe" age=29 sex=male
address="123 East NW Seattle"
```

Output:

```
None
```

2. Per elencare i tag di un cluster

--Comando:

```
aws emr describe-cluster --cluster-id j-XXXXXXYY --query Cluster.Tags
```

**Output:**

```
[
  {
    "Value": "male",
    "Key": "sex"
  },
  {
    "Value": "123 East NW Seattle",
    "Key": "address"
  },
  {
    "Value": "John Doe",
    "Key": "name"
  },
  {
    "Value": "29",
    "Key": "age"
  }
]
```

- Per API i dettagli, vedere [AddTags](#) in AWS CLI Command Reference.

**create-cluster-examples**

Il seguente esempio di codice mostra come utilizzare `create-cluster-examples`.

**AWS CLI**

La maggior parte degli esempi seguenti presuppone che tu abbia specificato il ruolo del EMR servizio Amazon e il profilo dell'EC2 istanza Amazon. Se non l'hai fatto, devi specificare ogni IAM ruolo richiesto o utilizzare il `--use-default-roles` parametro durante la creazione del cluster. Per ulteriori informazioni sulla specificazione IAM dei ruoli, consulta [Configure IAM Roles for Amazon EMR Permissions to AWS Services](#) nella Amazon EMR Management Guide.

**Esempio 1: creare un cluster**

L'`create-cluster` esempio seguente crea un EMR cluster semplice.

```
aws emr create-cluster \  
  --release-label emr-5.14.0 \  
  --
```



```
--instance-type m4.large \  
--instance-count 2
```

Questo comando non produce alcun output.

Esempio 2: creare un EMR cluster Amazon con impostazioni predefinite ServiceRole e InstanceProfile ruoli

L'`create-cluster` esempio seguente crea un EMR cluster Amazon che utilizza la `--instance-groups` configurazione.

```
aws emr create-cluster \  
  --release-label emr-5.14.0 \  
  --service-role EMR_DefaultRole \  
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \  
  --instance-  
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
```

Esempio 3: creare un EMR cluster Amazon che utilizza una flotta di istanze

L'`create-cluster` esempio seguente crea un EMR cluster Amazon che utilizza la `--instance-fleets` configurazione, specificando due tipi di istanze per ogni flotta e due EC2 sottoreti.

```
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
```

Esempio 4: creare un cluster con ruoli predefiniti

L'`create-cluster` esempio seguente utilizza il `--use-default-roles` parametro per specificare il ruolo di servizio e il profilo di istanza predefiniti.

```
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

```

Esempio 5: creare un cluster e specificare le applicazioni da installare

L'create-cluster esempio seguente utilizza il --applications parametro per specificare le applicazioni EMR installate da Amazon. Questo esempio installa Hadoop, Hive e Pig.

```

aws emr create-cluster \
    --applications Name=Hadoop Name=Hive Name=Pig \
    --release-label emr-5.9.0 \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE
\
    --auto-terminate

```

Esempio 6: creare un cluster che includa Spark

L'esempio seguente installa Spark.

```

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

```

Esempio 7: specificare una personalizzazione da utilizzare AMI per le istanze del cluster

L'create-cluster esempio seguente crea un'istanza di cluster basata su Amazon Linux AMI con IDami-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 \

```



```
--auto-terminate
```

L'esempio seguente fa riferimento `configurations.json` a un file 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
```

Esempio 9: creare un cluster con gruppi di istanze master, core e task

L'`create-cluster` esempio seguente specifica `--instance-groups` il tipo e il numero di EC2 istanze da utilizzare per i gruppi di istanze master, core e task.

```
aws emr create-cluster \
  --release-label emr-5.9.0 \
  --instance-
groups Name=Master,InstanceGroupType=MASTER,InstanceType=m4.Large,InstanceCount=1 Name=Core,
```

Esempio 10: specificare che un cluster deve terminare dopo aver completato tutti i passaggi

L'`create-cluster` esempio seguente specifica che il cluster deve chiudersi automaticamente dopo aver completato tutti i passaggi. `--auto-terminate`

```
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
```

Esempio 11: specificare i dettagli della configurazione del cluster come la coppia di EC2 chiavi Amazon, la configurazione di rete e i gruppi di sicurezza

L'`create-cluster` esempio seguente crea un cluster con la coppia di EC2 chiavi Amazon denominata `myKey` e un profilo di istanza personalizzato denominato `myProfile`. Le coppie di chiavi vengono utilizzate per autorizzare le SSH connessioni ai nodi del cluster, molto spesso al nodo master. Per ulteriori informazioni, consulta [Use an Amazon EC2 Key Pair for SSH Credentials](#) nella 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
```

L'esempio seguente crea un cluster in una VPC sottorete Amazon.

```
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
```

L'esempio seguente crea un cluster nella zona di us-east-1b disponibilità.

```
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
```

L'esempio seguente crea un cluster e specifica solo i gruppi di sicurezza EMR gestiti da Amazon.

```
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
```

L'esempio seguente crea un cluster e specifica solo gruppi di EC2 sicurezza Amazon aggiuntivi.

```
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

```

L'esempio seguente crea un cluster e specifica i gruppi di sicurezza EMR -Managed, nonché i gruppi di sicurezza aggiuntivi.

```

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

```

L'esempio seguente crea un cluster in una sottorete VPC privata e utilizza un gruppo di EC2 sicurezza Amazon specifico per abilitare l'accesso ai EMR servizi Amazon, necessario per i cluster in sottoreti private.

```

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

```

L'esempio seguente specifica i parametri di configurazione del gruppo di sicurezza utilizzando un JSON file denominato `ec2_attributes.json` archiviato localmente. NOTE: JSON gli argomenti devono includere opzioni e valori come elementi propri nell'elenco.

```

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

```

Contenuto di `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"]
  }
]

```

Esempio 12: per abilitare il debug e specificare un registro URI

L'`create-cluster` esempio seguente utilizza il `--enable-debugging` parametro, che consente di visualizzare i file di registro più facilmente utilizzando lo strumento di debug nella console Amazon. EMR il `--log-uri` parametro è obbligatorio con `--enable-debugging`

```

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

```

Esempio 13: per aggiungere tag durante la creazione di un cluster

I tag sono coppie chiave-valore che aiutano a identificare e gestire i cluster. L'`create-cluster` esempio seguente utilizza il `--tags` parametro per creare tre tag per un cluster,

uno con il nome della chiave `name` e il valore `Shirley Rodriguez`, un secondo con il nome della chiave `age` e il valore `29` e un terzo tag con il nome della chiave `department` e il valore `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
```

L'esempio seguente elenca i tag applicati a un cluster.

```
aws emr describe-cluster \  
  --cluster-id j-XXXXXXYY \  
  --query Cluster.Tags
```

Esempio 14: Utilizzare una configurazione di sicurezza che abiliti la crittografia e altre funzionalità di sicurezza

L'`create-cluster` esempio seguente utilizza il `--security-configuration` parametro per specificare una configurazione di sicurezza per un EMR cluster. Puoi utilizzare configurazioni di sicurezza con Amazon EMR versione 4.8.0 o successiva.

```
aws emr create-cluster \  
  --instance-type m4.large \  
  --release-label emr-5.9.0 \  
  --security-configuration mySecurityConfiguration
```

Esempio 15: creare un cluster con volumi di EBS storage aggiuntivi configurati per i gruppi di istanze

Quando si specificano EBS volumi aggiuntivi, sono richiesti i seguenti argomenti: `VolumeType`, `SizeInGB` se `EbsBlockDeviceConfigs` specificato.

L'`create-cluster` esempio seguente crea un cluster con più EBS volumi collegati alle EC2 istanze del gruppo di istanze principale.

```
aws emr create-cluster \  
  --security-configuration mySecurityConfiguration
```



```

--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

```

L'esempio seguente crea un cluster con più EBS volumi collegati alle EC2 istanze del gruppo di istanze principale.

```

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

```

Esempio 16: creare un cluster con una politica di scalabilità automatica

Puoi collegare politiche di scalabilità automatica ai gruppi di istanze principali e task utilizzando Amazon EMR versione 4.0 e successive. La policy di scalabilità automatica aggiunge e rimuove dinamicamente EC2 le istanze in risposta a un parametro Amazon. CloudWatch Per ulteriori informazioni, consulta Using Automatic Scaling in Amazon EMR < <https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-automatic-scaling.html> >`\_ nella Amazon Management Guide. EMR

Quando allegghi una politica di ridimensionamento automatico, devi anche specificare il ruolo predefinito per l'utilizzo del ridimensionamento automatico. `--auto-scaling-role EMR_AutoScaling_DefaultRole`

L'`create-cluster` esempio seguente specifica la politica di scalabilità automatica per il gruppo di CORE istanze utilizzando l'`AutoScalingPolicy` argomento con una JSON struttura incorporata, che specifica la configurazione della politica di scalabilità. I gruppi di istanze con una JSON struttura incorporata devono avere l'intera raccolta di argomenti racchiusa tra virgolette

single. L'uso delle virgolette singole è facoltativo, ad esempio per i gruppi senza una JSON struttura incorporata.

```
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
```

L'esempio seguente utilizza un JSON file per specificare la configurazione di tutti i gruppi di istanze in un cluster. `instancegroupconfig.json` Il JSON file specifica la configurazione della politica di scalabilità automatica per il gruppo di istanze principale.

```
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
```

Contenuto di `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}"
                    }
                ]
            }
        }
    ]
}
]

```

### Esempio 17: aggiungi JAR passaggi personalizzati durante la creazione di un cluster

L'create-cluster esempio seguente aggiunge passaggi specificando un JAR file archiviato in Amazon S3. Steps invia il lavoro a un cluster. La funzione principale definita nel JAR file viene eseguita dopo il provisioning EC2 delle istanze, l'esecuzione di eventuali azioni di bootstrap e l'installazione delle applicazioni. I passaggi vengono specificati utilizzando. Type=CUSTOM\_JAR

JAR I passaggi personalizzati richiedono il Jar= parametro, che specifica il percorso e il nome del JAR file di. I parametri opzionali sono TypeName, ActionOnFailure, Args, e MainClass. Se la

classe principale non è specificata, il JAR file deve essere specificato Main-Class nel relativo file manifest.

```
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
```

Esempio 18: Per aggiungere passaggi di streaming durante la creazione di un cluster

`create-cluster` Gli esempi seguenti aggiungono una fase di streaming a un cluster che termina dopo l'esecuzione di tutti i passaggi. Le fasi di streaming richiedono parametri `Type` e `Args`. I parametri opzionali delle fasi di streaming sono `Name` e `ActionOnFailure`.

L'esempio seguente specifica il passaggio inline.

```
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
```

L'esempio seguente utilizza un file di JSON configurazione memorizzato localmente denominato `multiplefiles.json`. La JSON configurazione specifica più file. Per specificare più file all'interno di un passaggio, è necessario utilizzare un file JSON di configurazione per specificare il passaggio. JSON gli argomenti devono includere opzioni e valori come elementi propri nell'elenco.

```
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

```

Contenuto di `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"
  }
]

```

Esempio 19: Per aggiungere passaggi Hive durante la creazione di un cluster

L'esempio seguente aggiunge i passaggi Hive durante la creazione di un cluster. I passaggi Hive richiedono parametri `Type` e `Args`. I parametri opzionali di Hive steps sono `Name` e `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
```

Esempio 20: aggiungere passaggi Pig durante la creazione di un cluster

L'esempio seguente aggiunge i passaggi Pig durante la creazione di un cluster. I parametri richiesti da Pig steps sono Type e. Args I parametri opzionali di Pig Steps sono Name e. 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
```

Esempio 21: Per aggiungere azioni bootstrap

L'create-cluster esempio seguente esegue due azioni di bootstrap definite come script archiviati in Amazon S3.

```
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
```

Esempio 22: per abilitare una visualizzazione EMRFS coerente e personalizzare le impostazioni e RetryCount RetryPeriod

L'create-cluster esempio seguente specifica il numero di tentativi e il periodo di tentativi per EMRFS una visualizzazione coerente. L'argomento Consistent=true è obbligatorio.

```
aws emr create-cluster \
  --instance-type m4.Large \
```

```
--release-label emr-5.9.0 \  
--emrfs Consistent=true,RetryCount=6,RetryPeriod=30
```

L'esempio seguente specifica la stessa EMRFS configurazione dell'esempio precedente, utilizzando un file di configurazione memorizzato JSON localmente denominato.

`emrfsconfig.json`

```
aws emr create-cluster \  
  --instance-type m4.large \  
  --release-label emr-5.9.0 \  
  --emrfs file://emrfsconfig.json
```

Contenuto di `emrfsconfig.json`.

```
{  
  "Consistent": true,  
  "RetryCount": 6,  
  "RetryPeriod": 30  
}
```

Esempio 23: creare un cluster con Kerberos configurato

`create-cluster` Gli esempi seguenti creano un cluster utilizzando una configurazione di sicurezza con Kerberos abilitato e stabiliscono i parametri Kerberos per il cluster che utilizza. --  
`kerberos-attributes`

Il comando seguente specifica gli attributi Kerberos per il cluster in linea.

```
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
```

Il comando seguente specifica gli stessi attributi, ma fa riferimento a un file memorizzato localmente denominato. `JSON kerberos_attributes.json` In questo esempio, il file viene salvato nella stessa directory in cui viene eseguito il comando. Puoi anche fare riferimento a un file di configurazione salvato in Amazon S3.

```
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
```

Contenuto di `kerberos_attributes.json`.

```
{
  "Realm": "EC2.INTERNAL",
  "KdcAdminPassword": "123",
  "CrossRealmTrustPrincipalPassword": "123",
}
```

L'`create-cluster` esempio seguente crea un EMR cluster Amazon che utilizza la `--instance-groups` configurazione e dispone di una politica di scalabilità gestita.

```
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}'
```

L'`create-cluster` esempio seguente crea un EMR cluster Amazon che utilizza «`--log-encryption-kms-key-id`» per definire l'ID della KMS chiave utilizzato per la crittografia dei log.

```
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
```

L'`create-cluster` esempio seguente crea un EMR cluster Amazon che utilizza la configurazione «`--placement-group-configs`» per posizionare i nodi master in un cluster ad alta



disponibilità (HA) all'interno di un gruppo di EC2 collocamento utilizzando la strategia di SPREAD posizionamento.

```
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.largeInstanceGroupType=CORE,
\
  --placement-group-configs InstanceRole=MASTER
```

L'`create-cluster` esempio seguente crea un EMR cluster Amazon che utilizza la configurazione «`--auto-termination-policy`» per impostare una soglia di terminazione automatica dei periodi di inattività per il cluster.

```
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
```

L'`create-cluster` esempio seguente crea un EMR cluster Amazon che utilizza il «`--os-release-label`» per definire una versione di Amazon Linux per il lancio del cluster

```
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,
```

Esempio 24: Per specificare gli attributi di un volume EBS root: dimensione, iops e velocità effettiva per le istanze di cluster create con le EMR versioni 6.15.0 e successive

L'`create-cluster` esempio seguente crea un EMR cluster Amazon che utilizza gli attributi del volume root per configurare le specifiche dei volumi root per le EC2 istanze.

```
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
```

- Per API i dettagli, consulta [CreateClusterExamples AWS CLI Command Reference](#).

## create-default-roles

Il seguente esempio di codice mostra come utilizzare `create-default-roles`.

### AWS CLI

1. Per creare il IAM ruolo predefinito per EC2

Comando:

```
aws emr create-default-roles
```

Output:

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": "AROAIQ5SIQUGL5KMYBJX6",
    "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"
}
]

```

- Per API i dettagli, vedere [CreateDefaultRoles](#) in AWS CLI Command Reference.

## create-security-configuration

Il seguente esempio di codice mostra come utilizzare `create-security-configuration`.

### AWS CLI

1. Per creare una configurazione di sicurezza con la crittografia in transito abilitata PEM per il fornitore di certificati e la crittografia a riposo abilitata con SSE -S3 per la crittografia S3 e AWS-KMS per il provider di chiavi del disco locale

**Comando:**

```
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"
      }
    }
  }
}'
```

**Output:**

```
{
  "CreationDateTime": 1474070889.129,
  "Name": "MySecurityConfig"
}
```

**JSONequivalente (contenuto di 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"
    }
}
}
}
}

```

Comando (utilizzando `security_configuration.json`):

```
aws emr create-security-configuration --name "MySecurityConfig" --security-
configuration file://./security_configuration.json
```

Output:

```
{
  "CreationDateTime": 1474070889.129,
  "Name": "MySecurityConfig"
}
```

2. Creare una configurazione di sicurezza con Kerberos abilitato utilizzando un trust dedicato al cluster KDC e cross-realm

Comando:

```
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"
    }
}
}'

```

Output:

```

{
  "CreationDateTime": 1490225558.982,
  "Name": "MySecurityConfig"
}

```

JSONequivalente (contenuto di 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"
        }
      }
    }
  }
}

```

Comando (utilizzando security\_configuration.json):

```

aws emr create-security-configuration --name "MySecurityConfig" --security-
configuration file:///./security_configuration.json

```

Output:



```
{
  "CreationDateTime": 1490225558.982,
  "Name": "MySecurityConfig"
}
```

- Per API i dettagli, vedere in Command Reference. [CreateSecurityConfiguration](#) AWS CLI

## delete-security-configuration

Il seguente esempio di codice mostra come utilizzare `delete-security-configuration`.

### AWS CLI

Per eliminare una configurazione di sicurezza nell'area corrente

Comando:

```
aws emr delete-security-configuration --name MySecurityConfig
```

Output:

```
None
```

- Per API i dettagli, vedere [DeleteSecurityConfiguration](#) in AWS CLI Command Reference.

## describe-cluster

Il seguente esempio di codice mostra come utilizzare `describe-cluster`.

### AWS CLI

Comando:

```
aws emr describe-cluster --cluster-id j-XXXXXXXX
```

Output:

```
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"
}
}

```

- Per API i dettagli, vedere [DescribeCluster](#) in AWS CLI Command Reference.

## describe-step

Il seguente esempio di codice mostra come utilizzare `describe-step`.

### AWS CLI

Il comando seguente descrive un passaggio con l'ID del passaggio `s-3LZC0QUT43AM` in un cluster con l'ID del cluster `j-3SD91U2E1L2QX`:

```
aws emr describe-step --cluster-id j-3SD91U2E1L2QX --step-id s-3LZC0QUT43AM
```

Output:

```
{
  "Step": {
    "Status": {
      "Timeline": {
        "EndDateTime": 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"
  }
}
```

- Per API i dettagli, vedere [DescribeStep](#) in AWS CLI Command Reference.

## get

Il seguente esempio di codice mostra come utilizzare `get`.



## AWS CLI

Quanto segue scarica l'`hadoop-examples.jar` archivio dall'istanza master in un cluster con l'ID del cluster `j-3SD91U2E1L2QX`:

```
aws emr get --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src /home/hadoop-examples.jar --dest ~
```

- Per API i dettagli, consulta [Get](#) in AWS CLI Command Reference.

## list-clusters

Il seguente esempio di codice mostra come usare `list-clusters`.

## AWS CLI

Il comando seguente elenca tutti i EMR cluster attivi nella regione corrente:

```
aws emr list-clusters --active
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ListClusters](#) in AWS CLI Command Reference.

## list-instance-fleets

Il seguente esempio di codice mostra come utilizzare `list-instance-fleets`.

### AWS CLI

Per ottenere i dettagli di configurazione delle flotte di istanze in un cluster

Questo esempio elenca i dettagli delle flotte di istanze nel cluster specificato.

Comando:

```
list-instance-fleets --cluster-id 'j-12ABCDEFGHI34JK'
```

Output:

```
{
  "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"
}
]
}

```

- Per API i dettagli, vedere [ListInstanceFleets](#) in AWS CLI Command Reference.

## list-instances

Il seguente esempio di codice mostra come utilizzare `list-instances`.

### AWS CLI

Il comando seguente elenca tutte le istanze di un cluster con l'ID cluster: `j-3C6XNQ39VR9WL`

```
aws emr list-instances --cluster-id j-3C6XNQ39VR9WL
```

## Output:

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-3NNHQ2TWB6Y",
      "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"
    }
  ]
}

```

```
}
```

- Per API i dettagli, vedere [ListInstances](#) in AWS CLI Command Reference.

## list-security-configurations

Il seguente esempio di codice mostra come utilizzare `list-security-configurations`.

### AWS CLI

Per elencare le configurazioni di sicurezza nell'area corrente

Comando:

```
aws emr list-security-configurations
```

Output:

```
{
  "SecurityConfigurations": [
    {
      "CreationDateTime": 1473889697.417,
      "Name": "MySecurityConfig-1"
    },
    {
      "CreationDateTime": 1473889697.417,
      "Name": "MySecurityConfig-2"
    }
  ]
}
```

- Per API i dettagli, vedere [ListSecurityConfigurations](#) in AWS CLI Command Reference.

## list-steps

Il seguente esempio di codice mostra come utilizzare `list-steps`.

### AWS CLI

Il comando seguente elenca tutti i passaggi di un cluster con l'ID cluster `j-3SD91U2E1L2QX`:

```
aws emr list-steps --cluster-id j-3SD91U2E1L2QX
```

- Per API i dettagli, vedere [ListSteps](#) in AWS CLI Command Reference.

## modify-cluster-attributes

Il seguente esempio di codice mostra come utilizzare `modify-cluster-attributes`.

### AWS CLI

Il comando seguente imposta la visibilità di un EMR cluster con l'ID `j-301CDNY0J5XM4` per tutti gli utenti:

```
aws emr modify-cluster-attributes --cluster-id j-301CDNY0J5XM4 --visible-to-all-users
```

- Per API i dettagli, vedere [ModifyClusterAttributes](#) in AWS CLI Command Reference.

## modify-instance-fleet

Il seguente esempio di codice mostra come utilizzare `modify-instance-fleet`.

### AWS CLI

Per modificare le capacità target di una flotta di istanze

Questo esempio modifica le capacità target On-Demand e Spot a 1 per il parco istanze specificato.

Comando:

```
aws emr modify-instance-fleet --cluster-id 'j-12ABCDEFGHI34JK' --instance-fleet InstanceFleetId='if-2ABC4DEFGHIJ4',TargetOnDemandCapacity=1,TargetSpotCapacity=1
```

- Per API i dettagli, vedere [ModifyInstanceFleet](#) in AWS CLI Command Reference.

## put

Il seguente esempio di codice mostra come utilizzare `put`.

## AWS CLI

Il comando seguente carica un file denominato `healthcheck.sh` all'istanza master in un cluster con l'ID del cluster: `j-3SD91U2E1L2QX`

```
aws emr put --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src ~/scripts/healthcheck.sh --dest /home/hadoop/bin/healthcheck.sh
```

- Per API i dettagli, consulta [Put](#) in AWS CLI Command Reference.

## remove-tags

Il seguente esempio di codice mostra come utilizzare `remove-tags`.

## AWS CLI

Il comando seguente rimuove un tag con la chiave `prod` da un cluster con l'ID del cluster `j-3SD91U2E1L2QX`:

```
aws emr remove-tags --resource-id j-3SD91U2E1L2QX --tag-keys prod
```

- Per API i dettagli, vedere [RemoveTags](#) in AWS CLI Command Reference.

## schedule-hbase-backup

Il seguente esempio di codice mostra come utilizzare `schedule-hbase-backup`.

## AWS CLI

Nota: questo comando può essere utilizzato solo con le HBase AMI versioni 2.x e 3.x

1. Per pianificare un HBase backup completo >>>>>>  
`06ab6d6e13564b5733d75abaf3b599f93cf39a23`

Comando:

```
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
```

Output:



None

## 2. Per pianificare un backup incrementale HBase

Comando:

```
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
```

Output:

None

- Per API i dettagli, vedere [ScheduleHbaseBackup](#) in AWS CLI Command Reference.

## socks

Il seguente esempio di codice mostra come utilizzare `socks`.

### AWS CLI

Il comando seguente apre una connessione `socks` con l'istanza master in un cluster con l'ID del cluster: `j-3SD91U2E1L2QX`

```
aws emr socks --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

L'opzione `key pair file` richiede un percorso locale a un file di chiave privata.

- Per API i dettagli, consulta [Socks](#) in AWS CLI Command Reference.

## ssh

Il seguente esempio di codice mostra come utilizzare `ssh`.

### AWS CLI

Il comando seguente apre una connessione `ssh` con l'istanza master in un cluster con l'ID del cluster: `j-3SD91U2E1L2QX`

```
aws emr ssh --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

L'opzione key pair file richiede un percorso locale a un file di chiave privata.

Output:

```
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 ~]$
```

- Per API i dettagli, vedere [Ssh](#) in AWS CLI Command Reference.

## Amazon EMR su EKS esempi di utilizzo AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con Amazon EMR onEKS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **update-role-trust-policy**

Il seguente esempio di codice mostra come utilizzare `update-role-trust-policy`.

AWS CLI

Per aggiornare la politica di fiducia di un IAM ruolo da utilizzare con Amazon EMR su EKS

Questo comando di esempio aggiorna la policy di trust di un ruolo denominato `example_iam_role` in modo che possa essere utilizzato con Amazon EMR on EKS con lo spazio dei nomi `example_namespace` da un cluster denominato `example_cluster`. EKS

Comando:

```
aws emr-containers update-role-trust-policy \  
  --cluster example_cluster \  
  --namespace example_namespace \  
  --role-name example_iam_role
```

Output:

```
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.

- API Per [UpdateRoleTrustPolicy AWS CLI](#) i dettagli, consulta Command Reference.

## EventBridge esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with EventBridge.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **delete-rule**

Il seguente esempio di codice mostra come utilizzare `delete-rule`.

#### AWS CLI

Per eliminare una regola CloudWatch Events

Questo esempio elimina la regola denominata `EC2InstanceStateChanges`:

```
aws events delete-rule --name "EC2InstanceStateChanges"
```

- Per API i dettagli, vedere [DeleteRule](#) in AWS CLI Command Reference.

#### **describe-rule**

Il seguente esempio di codice mostra come utilizzare `describe-rule`.

## AWS CLI

Per visualizzare informazioni su una regola CloudWatch Events

Questo esempio visualizza informazioni sulla regola denominata `DailyLambdaFunction`:

```
aws events describe-rule --name "DailyLambdaFunction"
```

- Per API i dettagli, vedere [DescribeRule](#) in AWS CLI Command Reference.

## **disable-rule**

Il seguente esempio di codice mostra come utilizzare `disable-rule`.

## AWS CLI

Per disabilitare una regola CloudWatch Events

Questo esempio disabilita la regola denominata `DailyLambdaFunction`. La regola non viene eliminata:

```
aws events disable-rule --name "DailyLambdaFunction"
```

- Per API i dettagli, vedere [DisableRule](#) in AWS CLI Command Reference.

## **enable-rule**

Il seguente esempio di codice mostra come utilizzare `enable-rule`.

## AWS CLI

Per abilitare una regola CloudWatch Events

Questo esempio abilita la regola denominata `DailyLambdaFunction`, che era stata precedentemente disabilitata:

```
aws events enable-rule --name "DailyLambdaFunction"
```

- Per API i dettagli, vedere [EnableRule](#) in AWS CLI Command Reference.

## **list-rule-names-by-target**

Il seguente esempio di codice mostra come utilizzare `list-rule-names-by-target`.

### AWS CLI

Per visualizzare tutte le regole che hanno un obiettivo specificato

Questo esempio visualizza tutte le regole che hanno come destinazione la funzione Lambda denominata `MyFunctionName`:

```
aws events list-rule-names-by-target --target-arn "arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

- Per API i dettagli, consulta [ListRuleNamesByTarget AWS CLI Command Reference](#).

## **list-rules**

Il seguente esempio di codice mostra come utilizzare `list-rules`.

### AWS CLI

Per visualizzare un elenco di tutte le regole CloudWatch degli eventi

Questo esempio visualizza tutte le regole CloudWatch Events della regione:

```
aws events list-rules
```

Per visualizzare un elenco di regole CloudWatch Events che iniziano con una determinata stringa.

Questo esempio visualizza tutte le regole CloudWatch Events nella regione il cui nome inizia con «Daily»:

```
aws events list-rules --name-prefix "Daily"
```

- Per API i dettagli, vedere [ListRules](#) in AWS CLI Command Reference.

## **list-targets-by-rule**

Il seguente esempio di codice mostra come utilizzare `list-targets-by-rule`.

## AWS CLI

Per visualizzare tutti gli obiettivi di una regola CloudWatch Events

Questo esempio visualizza tutti gli obiettivi della regola denominata DailyLambdaFunction:

```
aws events list-targets-by-rule --rule "DailyLambdaFunction"
```

- Per API i dettagli, vedere [ListTargetsByRule](#) in AWS CLI Command Reference.

## put-events

Il seguente esempio di codice mostra come utilizzare `put-events`.

## AWS CLI

Per inviare un evento personalizzato a CloudWatch Events

Questo esempio invia un evento personalizzato a CloudWatch Events. L'evento è contenuto nel file `putevents.json`:

```
aws events put-events --entries file://putevents.json
```

Visualizzare il contenuto del file `putevents.json`:

```
[
  {
    "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"
  }
]

```

- Per API i dettagli, vedere [PutEvents](#) in AWS CLI Command Reference.

## put-rule

Il seguente esempio di codice mostra come utilizzare `put-rule`.

### AWS CLI

Per creare regole relative CloudWatch agli eventi

Questo esempio crea una regola che viene attivata ogni giorno alle 9:00am ()UTC. Se usi `put-targets` per aggiungere una funzione Lambda come destinazione di questa regola, puoi eseguire la funzione Lambda ogni giorno all'ora specificata:

```
aws events put-rule --name "DailyLambdaFunction" --schedule-expression "cron(0 9 * * ? *)"
```

Questo esempio crea una regola che si attiva quando un'EC2istanza nella regione cambia stato:

```
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"
```

Questo esempio crea una regola che si attiva quando un'EC2istanza nella regione viene interrotta o terminata:

```
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"
```

- Per API i dettagli, vedere [PutRule](#) in AWS CLI Command Reference.

## put-targets

Il seguente esempio di codice mostra come utilizzare `put-targets`.



## AWS CLI

Per aggiungere obiettivi per le regole CloudWatch Events

Nell'esempio seguente viene aggiunta una funzione Lambda come destinazione di una regola:

```
aws events put-targets --rule DailyLambdaFunction --targets  
  "Id"="1", "Arn"="arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

Questo esempio imposta un flusso Amazon Kinesis come destinazione, in modo che gli eventi rilevati da questa regola vengano inoltrati allo stream:

```
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"
```

Questo esempio imposta due flussi Amazon Kinesis come destinazione per una regola:

```
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"
```

- Per API i dettagli, vedere [PutTargets](#) in AWS CLI Command Reference.

## remove-targets

Il seguente esempio di codice mostra come utilizzare `remove-targets`.

## AWS CLI

Per rimuovere una destinazione per un evento

Questo esempio rimuove lo stream Amazon Kinesis denominato MyStream 1 dall'obiettivo della regola. DailyLambdaFunction Quando DailyLambdaFunction è stato creato, questo flusso è stato impostato come destinazione con un ID Target1:

```
aws events remove-targets --rule "DailyLambdaFunction" --ids "Target1"
```

- Per API i dettagli, consulta AWS CLI Command [RemoveTargets](#) Reference.

## test-event-pattern

Il seguente esempio di codice mostra come utilizzare `test-event-pattern`.

### AWS CLI

Per verificare se un modello di evento corrisponde a un evento specificato

Questo esempio verifica se il pattern «source:com.mycompany.myapp» corrisponde all'evento specificato. In questo esempio, l'output sarebbe «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\"}"
```

- Per API i dettagli, vedere [TestEventPattern](#) in AWS CLI Command Reference.

## Esempi di Firewall Manager che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface con Firewall Manager.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni illustrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### associate-admin-account

Il seguente esempio di codice mostra come utilizzare `associate-admin-account`.

## AWS CLI

Per impostare l'account amministratore di Firewall Manager

L'associate-admin-accountesempio seguente imposta l'account amministratore per Firewall Manager.

```
aws fms associate-admin-account \  
  --admin-account 123456789012
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [la sezione Impostazione dell'account amministratore di AWSAWS Firewall](#) Manager nella AWS WAF Guida per sviluppatori avanzata di Firewall Manager and AWS Shield.

- Per API i dettagli, vedere [AssociateAdminAccount](#) in AWS CLI Command Reference.

## delete-notification-channel

Il seguente esempio di codice mostra come utilizzaredelete-notification-channel.

### AWS CLI

Per rimuovere le informazioni sull'SNSargomento per i registri di Firewall Manager

L'delete-notification-channelesempio seguente rimuove le informazioni SNS sull'argomento.

```
aws fms delete-notification-channel
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configure Amazon SNS Notifications and Amazon CloudWatch Alarms](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [DeleteNotificationChannel AWS CLI](#) Command Reference.

## delete-policy

Il seguente esempio di codice mostra come utilizzaredelete-policy.

## AWS CLI

Per eliminare una politica di Firewall Manager

L'`delete-policy` esempio seguente rimuove la policy con l'ID specificato, insieme a tutte le relative risorse.

```
aws fms delete-policy \  
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --delete-all-policy-resources
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Working with AWS Firewall Manager Policies](#) nella AWS WAF Guida per sviluppatori avanzata di AWS Firewall Manager and AWS Shield.

- Per API i dettagli, vedere [DeletePolicy](#) in AWS CLI Command Reference.

## **disassociate-admin-account**

Il seguente esempio di codice mostra come utilizzare `disassociate-admin-account`.

### AWS CLI

Per rimuovere l'account amministratore di Firewall Manager

L'`disassociate-admin-account` esempio seguente rimuove l'associazione corrente degli account di amministratore da Firewall Manager.

```
aws fms disassociate-admin-account
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [la sezione Impostazione dell'account amministratore di AWSAWS Firewall](#) Manager nella AWS WAF Guida per sviluppatori avanzata di Firewall Manager and AWS Shield.

- Per API i dettagli, vedere [DisassociateAdminAccount](#) in AWS CLI Command Reference.

## **get-admin-account**

Il seguente esempio di codice mostra come utilizzare `get-admin-account`.

## AWS CLI

Per recuperare l'account amministratore di Firewall Manager

L'`get-admin-account` seguente recupera l'account dell'amministratore.

```
aws fms get-admin-account
```

Output:

```
{
  "AdminAccount": "123456789012",
  "RoleStatus": "READY"
}
```

Per ulteriori informazioni, consulta la sezione [Prerequisiti di AWS Firewall Manager](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [GetAdminAccount](#) in AWS CLI Command Reference.

## `get-compliance-detail`

Il seguente esempio di codice mostra come utilizzare `get-compliance-detail`.

## AWS CLI

Per recuperare le informazioni sulla conformità di un account

L'`get-compliance-detail` seguente recupera le informazioni sulla conformità per la politica e l'account membro specificati.

```
aws fms get-compliance-detail \
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --member-account 123456789012
```

Output:

```
{
  "PolicyComplianceDetail": {
    "EvaluationLimitExceeded": false,
    "IssueInfoMap": {},
    "MemberAccount": "123456789012",
```

```
"PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
"PolicyOwner": "123456789012",  
"Violators": []  
}
```

Per ulteriori informazioni, vedere [Visualizzazione della conformità delle risorse con una policy](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [GetComplianceDetail](#) in AWS CLI Command Reference.

## get-notification-channel

Il seguente esempio di codice mostra come utilizzare `get-notification-channel`.

### AWS CLI

Per recuperare le informazioni sull'SNS argomento per i log di Firewall Manager

L'`get-notification-channel` esempio seguente recupera le informazioni sull'SNS argomento.

```
aws fms get-notification-channel
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Configure Amazon SNS Notifications and Amazon CloudWatch Alarms](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [GetNotificationChannel AWS CLI](#) Command Reference.

## get-policy

Il seguente esempio di codice mostra come utilizzare `get-policy`.

### AWS CLI

Per recuperare una policy di Firewall Manager

L'`get-policy` seguente recupera la policy con l'ID specificato.

```
aws fms get-policy \
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, vedere [Working with AWS Firewall Manager Policies](#) nella AWS WAF Guida per sviluppatori avanzata di AWS Firewall Manager and AWS Shield.

- Per API i dettagli, vedere [GetPolicy](#) in AWS CLI Command Reference.

## **list-compliance-status**

Il seguente esempio di codice mostra come utilizzare `list-compliance-status`.

AWS CLI

Per recuperare le informazioni sulla conformità delle politiche per gli account dei membri

L'`list-compliance-status` seguente recupera le informazioni sulla conformità degli account membri per la politica specificata.

```
aws fms list-compliance-status \  
  --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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": {}  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Visualizzazione della conformità delle risorse con una policy](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [ListComplianceStatus](#) in AWS CLI Command Reference.

## list-member-accounts

Il seguente esempio di codice mostra come utilizzare `list-member-accounts`.

### AWS CLI

Per recuperare gli account dei membri dell'organizzazione



L'`list-member-accounts` seguente elenca tutti gli account membri che fanno parte dell'organizzazione dell'amministratore di Firewall Manager.

```
aws fms list-member-accounts
```

Output:

```
{
  "MemberAccounts": [
    "222222222222",
    "333333333333",
    "444444444444"
  ]
}
```

Per ulteriori informazioni, consulta [AWS Firewall Manager](#) nella AWS WAF, AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [ListMemberAccounts](#) in AWS CLI Command Reference.

## **list-policies**

Il seguente esempio di codice mostra come utilizzare `list-policies`.

AWS CLI

Per recuperare tutte le policy di Firewall Manager

L'`list-policies` seguente recupera l'elenco delle politiche per l'account. In questo esempio, l'output è limitato a due risultati per richiesta. Ogni chiamata restituisce un valore `NextToken` che può essere utilizzato come valore per il `--starting-token` parametro nella `list-policies` chiamata successiva per ottenere il successivo set di risultati per l'elenco.

```
aws fms list-policies \
  --max-items 2
```

Output:

```
{
```

```

"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=="
}

```

Per ulteriori informazioni, vedere [Working with AWS Firewall Manager Policies](#) nella AWS WAF Guida per sviluppatori avanzata di AWS Firewall Manager and AWS Shield.

- Per API i dettagli, vedere [ListPolicies](#) in AWS CLI Command Reference.

## put-notification-channel

Il seguente esempio di codice mostra come utilizzare `put-notification-channel`.

### AWS CLI

Per impostare le informazioni sull'SNS argomento per i log di Firewall Manager

L'`put-notification-channel` esempio seguente imposta le informazioni sull'SNS argomento.

```

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

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configure Amazon SNS Notifications and Amazon CloudWatch Alarms](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [PutNotificationChannel AWS CLI Command Reference](#).

## put-policy

Il seguente esempio di codice mostra come utilizzare `put-policy`.

### AWS CLI

Per creare una policy Firewall Manager

L'`put-policy` esempio seguente crea una politica del gruppo di sicurezza Firewall Manager.

```
aws fms put-policy \  
  --cli-input-json file://policy.json
```

Contenuto di `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"  
    }  
  ]  
}
```

## Output:

```
{
  "Policy": {
    "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "PolicyName": "test",
    "PolicyUpdateToken": "1:X9QGexP7HASDlsFp+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"
}
```

Per ulteriori informazioni, vedere [Working with AWS Firewall Manager Policies](#) nella AWS WAF Guida per sviluppatori avanzata di AWS Firewall Manager and AWS Shield.

- Per API i dettagli, vedere [PutPolicy](#) in AWS CLI Command Reference.

## AWS FIS esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS FIS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **create-experiment-template**

Il seguente esempio di codice mostra come utilizzare `create-experiment-template`.

#### AWS CLI

Per creare un modello di esperimento

L'esempio seguente crea un modello di esperimento nel tuo AWS FIS account.

```
aws fis create-experiment-template \  
  --cli-input-json file://myfile.json
```

Contenuto di `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"  
}
```

### Output:

```
{  
  "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": {}  
  }  
}
```

Per ulteriori informazioni, consulta [Creare un modello di esperimento](#) nella Guida per l'utente di AWS Fault Injection Simulator.

- Per API i dettagli, vedere [CreateExperimentTemplate](#) in AWS CLI Command Reference.

## delete-experiment-template

Il seguente esempio di codice mostra come utilizzare `delete-experiment-template`.

### AWS CLI

Per eliminare un modello di esperimento

L'`delete-experiment-template` seguente elimina il modello di esperimento specificato.

```
aws fis delete-experiment-template \  
  --id ABCDE1fgHIJkLmNop
```

Output:

```
{  
  "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"
}
}

```

Per ulteriori informazioni, vedete [Eliminare un modello di esperimento](#) nella Guida per l'utente di AWS Fault Injection Simulator.

- Per API i dettagli, vedere [DeleteExperimentTemplate](#) in AWS CLI Command Reference.

## get-action

Il seguente esempio di codice mostra come utilizzare `get-action`.

### AWS CLI

Per ottenere i dettagli delle azioni

L'`get-action` esempio seguente ottiene i dettagli dell'azione specificata.

```

aws fis get-action \
  --id aws:ec2:stop-instances

```

Output:

```

{
  "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": {}
}
}

```

Per ulteriori informazioni, vedere [Azioni](#) nella guida per l'utente di AWS Fault Injection Simulator.

- Per API i dettagli, vedere [GetAction](#) in AWS CLI Command Reference.

## get-experiment-template

Il seguente esempio di codice mostra come utilizzare `get-experiment-template`.

### AWS CLI

Per ottenere i dettagli del modello di esperimento

L'`get-experiment-template` esempio seguente ottiene i dettagli del modello di esperimento specificato.

```
aws fis get-experiment-template \
  --id ABCDE1fgHIJkLmNop
```

Output:

```

{
  "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"
    }
  }
}

```

Per ulteriori informazioni, vedete [Modelli di esperimenti](#) nella Guida per l'utente di AWS Fault Injection Simulator.

- Per API i dettagli, vedere [GetExperimentTemplate](#) in AWS CLI Command Reference.

## get-experiment

Il seguente esempio di codice mostra come utilizzare `get-experiment`.

### AWS CLI

Per ottenere i dettagli dell'esperimento

L'`get-experiment` seguente ottiene i dettagli dell'esperimento specificato.

```
aws fis get-experiment \
  --id ABC12DeFGhI3jKLMNOP
```

## Output:

```
{
  "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": {}
  }
}
```

```
}
```

Per ulteriori informazioni, vedere [Experiments for AWS FIS](#) nella Guida per l'utente di AWS Fault Injection Simulator.

- Per API i dettagli, vedere [GetExperiment](#) in AWS CLI Command Reference.

## list-actions

Il seguente esempio di codice mostra come utilizzare `list-actions`.

### AWS CLI

Per elencare le azioni

L'`list-actions` esempio seguente elenca le azioni disponibili.

```
aws fis list-actions
```

Output:

```
{
  "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": {}
  }
]
}
```

Per ulteriori informazioni, vedere [Actions](#) in the AWS Fault Injection Simulator User Guide.

- Per API i dettagli, vedere [ListActions](#) in AWS CLI Command Reference.

## list-experiment-templates

Il seguente esempio di codice mostra come utilizzare `list-experiment-templates`.

### AWS CLI

Per elencare i modelli di esperimenti

L'`list-experiment-templates` esempio seguente elenca i modelli di esperimento presenti nel tuo AWS account.

```
aws fis list-experiment-templates
```

Output:

```
{
  "experimentTemplates": [
    {
      "id": "ABCDE1fgHIJKLmNop",
      "description": "myExperimentTemplate",
      "creationTime": 1616017191.124,
      "lastUpdateTime": 1616017191.124,
    }
  ]
}
```

```
        "tags": {
            "key": "value"
        }
    ]
}
```

Per ulteriori informazioni, consulta [Modelli di esperimenti](#) nella Guida per l'utente di AWS Fault Injection Simulator.

- Per API i dettagli, vedere [ListExperimentTemplates](#) in AWS CLI Command Reference.

## list-experiments

Il seguente esempio di codice mostra come utilizzare `list-experiments`.

### AWS CLI

Per elencare gli esperimenti

L'`list-experiments` esempio seguente elenca gli esperimenti nel tuo AWS account.

```
aws fis list-experiments
```

Output:

```
{
  "experiments": [
    {
      "id": "ABCdeF1GHiJkLM23N0",
      "experimentTemplateId": "ABCDE1fgHIJkLmNop",
      "state": {
        "status": "running",
        "reason": "Experiment is running."
      },
      "creationTime": 1616017341.197,
      "tags": {
        "key": "value"
      }
    }
  ]
}
```



Per ulteriori informazioni, consultate [Experiments](#) in the AWS Fault Injection Simulator User Guide.

- Per API i dettagli, vedere [ListExperiments](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una risorsa

L'`list-tags-for-resource` esempio seguente elenca i tag per la risorsa specificata.

```
aws fis list-tags-for-resource \  
  --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

Output:

```
{  
  "tags": {  
    "key1": "value1",  
    "key2": "value2"  
  }  
}
```

Per ulteriori informazioni, vedete [Etichettare AWS FIS le vostre risorse](#) nella Guida per l'utente di AWS Fault Injection Simulator.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## start-experiment

Il seguente esempio di codice mostra come utilizzare `start-experiment`.

### AWS CLI

Per iniziare un esperimento

L'`start-experiment` esempio seguente avvia l'esperimento specificato.

```
aws fis start-experiment \  
--experiment-template-id ABCDE1fgHIJkLmNop
```

Output:

```
{  
  "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": {}  
  }  
}
```

Per ulteriori informazioni, vedete [Experiments for AWS FIS](#) nella Guida per l'utente di AWS Fault Injection Simulator.

- Per API i dettagli, vedere [StartExperiment](#) in AWS CLI Command Reference.

## stop-experiment

Il seguente esempio di codice mostra come utilizzare `stop-experiment`.

### AWS CLI

Interrompere un esperimento

L'`stop-experiment` seguente interrompe l'esecuzione dell'esperimento specificato.

```
aws fis stop-experiment \  
  --id ABC12DeFGhI3jKLMNOP
```

Output:

```
{  
  "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": {}
}
}

```

Per ulteriori informazioni, vedete [Experiments for AWS FIS](#) nella Guida per l'utente di AWS Fault Injection Simulator.

- Per API i dettagli, vedere [StopExperiment](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare una risorsa

L'`tag-resource`esempio seguente contrassegna la risorsa specificata.

```
aws fis tag-resource \  
  --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP \  
  --tags key1=value1,key2=value2
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Etichettare le AWS FIS risorse](#) nella Guida per l'utente di AWS Fault Injection Simulator.

- Per API i dettagli, vedere [TagResource](#)in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una risorsa

L'`untag-resource`esempio seguente rimuove i tag dalla risorsa specificata.

```
aws fis untag-resource \  
  --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Etichettare AWS FIS le vostre risorse](#) nella Guida per l'utente di AWS Fault Injection Simulator.

- Per API i dettagli, vedere [UntagResource](#)in AWS CLI Command Reference.

## update-experiment-template

Il seguente esempio di codice mostra come utilizzare `update-experiment-template`.

### AWS CLI

Per aggiornare un modello di esperimento

L'esempio seguente aggiorna la descrizione del modello di esperimento specificato.

```
aws fis update-experiment-template \  
  --id ABCDE1fgHIJkLmNop \  
  ---description myExperimentTemplate
```

Output:

```
{  
  "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"
  }
}
}
```

Per ulteriori informazioni, vedete [Aggiornare un modello di esperimento](#) nella Guida per l'utente di AWS Fault Injection Simulator.

- Per API i dettagli, vedere [UpdateExperimentTemplate](#) in AWS CLI Command Reference.

## GameLift Esempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con Amazon GameLift.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-build**

Il seguente esempio di codice mostra come utilizzare `create-build`.

AWS CLI

Esempio 1: creare una build di gioco dai file in un bucket S3

L'`create-build` seguente crea una risorsa di build di gioco personalizzata. Utilizza file compressi che vengono archiviati in una posizione S3 in un AWS account che controlli tu. Questo esempio presuppone che tu abbia già creato un IAM ruolo che GameLift autorizza Amazon ad accedere alla posizione S3. Poiché la richiesta non specifica un sistema operativo, il valore predefinito della nuova risorsa di build è `_2012.WINDOWS`

```
aws gamelift create-build \  
  --storage-location file://storage-loc.json \  
  --name MegaFrogRaceServer.NA \  
  --build-version 12345.678
```

Contenuto di `storage-loc.json`.

```
{  
  "Bucket": "MegaFrogRaceServer_NA_build_files"  
  "Key": "MegaFrogRaceServer_build_123.zip"  
  "RoleArn": "arn:aws:iam::123456789012:role/gamelift"  
}
```

Output:

```
{  
  "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"  
  }  
}
```

Esempio 2: creare una risorsa di compilazione del gioco per caricare manualmente i file su GameLift



L'create-build esempio seguente crea una nuova risorsa di build. Inoltre, ottiene una posizione di archiviazione e credenziali temporanee che ti consentono di caricare manualmente la build del gioco nella GameLift posizione in Amazon S3. Dopo aver caricato correttamente la build, il GameLift servizio convalida la build e aggiorna lo stato della nuova build.

```
aws gamelift create-build \  
  --name MegaFrogRaceServer.NA \  
  --build-version 12345.678 \  
  --operating-system AMAZON_LINUX
```

Output:

```
{  
  "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": "AgoGb3JpZ221uENz...EXAMPLETOKEN=="  
  }  
}
```

Per ulteriori informazioni, consulta [Caricare una build di server personalizzata GameLift](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta [CreateBuild AWS CLI Command Reference](#).

## create-fleet

Il seguente esempio di codice mostra come utilizzare `create-fleet`.

### AWS CLI

Esempio 1: creare una flotta Linux di base

L'esempio seguente crea una flotta minimamente configurata di istanze Linux su richiesta per ospitare una build di server personalizzata. È possibile completare la configurazione utilizzando `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}]'
```

Output:

```
{  
  "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"  
  }  
}
```

```
}
}
```

## Esempio 2: creare una flotta Windows di base

L'create-fleetesempio seguente crea una flotta di istanze spot di Windows con configurazione minima per ospitare una build di server personalizzata. È possibile completare la configurazione utilizzando 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}]'
```

Output:

```
{
  "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"
  }
}
```

}

### Esempio 3: creare una flotta completamente configurata

L'`create-fleet` seguente crea una flotta di istanze Spot Windows per una build di server personalizzata, con le impostazioni di configurazione più comunemente utilizzate.

```
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'
```

Contenuto di `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}]
```

Output:

```
{
  "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"
    }
  }
}

```

#### Esempio 4: creare una flotta di server in tempo reale

L'`create-fleet` seguente crea una flotta di istanze Spot con uno script di configurazione in tempo reale che è stato caricato su Amazon. GameLift Tutti i server Realtime vengono distribuiti su macchine Linux. Ai fini di questo esempio, si supponga che lo script Realtime caricato includa più file di script, con la `Init()` funzione che si trova nel file di script chiamata `MainScript.js`. Come illustrato, questo file viene identificato come script di avvio nella configurazione di runtime.

```

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}]'

```

Output:

```

{
  "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"
  }
}

```

- Per API i dettagli, vedere [CreateFleet](#) in AWS CLI Command Reference.

## create-game-session-queue

Il seguente esempio di codice mostra come utilizzare `create-game-session-queue`.

### AWS CLI

Esempio 1: Per impostare una coda ordinata per le sessioni di gioco

L'`create-game-session-queue` esempio seguente crea una nuova coda per le sessioni di gioco con destinazioni in due regioni. Inoltre, configura la coda in modo che le richieste di sessione di gioco scadano dopo aver atteso 10 minuti per il posizionamento. Poiché non sono state definite politiche di latenza, GameLift tenta di collocare tutte le sessioni di gioco con la prima destinazione elencata.

```

aws gamelift create-game-session-queue \
  --name MegaFrogRaceServer-NA \
  --destinations file://destinations.json \
  --timeout-in-seconds 600

```

Contenuto di `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" }
  ]
}
```

### Output:

```
{
  "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"}
      ]
    }
  ]
}
```

Esempio 2: per impostare una coda per le sessioni di gioco con le politiche di latenza dei giocatori

L'create-game-session-queueesempio seguente crea una nuova coda di sessioni di gioco con politiche di latenza per due giocatori. La prima politica stabilisce un limite di latenza di 100 ms che viene applicato durante il primo minuto di un tentativo di posizionamento della sessione di gioco. La seconda politica innalza il limite di latenza a 200 ms fino al timeout della richiesta di posizionamento, pari a 3 minuti.

```
aws gamelift create-game-session-queue \
  --name MegaFrogRaceServer-NA \
  --destinations file://destinations.json \
  --player-latency-policies file://latency-policies.json \
  --timeout-in-seconds 180
```

## Contenuto di 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" }
  ]
}
```

## Contenuto di latency-policies.json.

```
{
  "PlayerLatencyPolicies": [
    {"MaximumIndividualPlayerLatencyMilliseconds": 200},
    {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":
60}
  ]
}
```

## Output:

```
{
  "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"}
  ]
}
```



```
    ],  
  }  
}
```

Per ulteriori informazioni, consulta [Create a Queue](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta [CreateGameSessionQueue AWS CLI Command Reference](#).

## delete-build

Il seguente esempio di codice mostra come utilizzare `delete-build`.

### AWS CLI

Per eliminare una build di gioco personalizzata

L'esempio seguente rimuove una build dal tuo GameLift account Amazon. Dopo l'eliminazione della build, non puoi utilizzarla per creare nuove flotte. Questa operazione non può essere annullata.

```
aws gamelift delete-build \  
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteBuild](#) in AWS CLI Command Reference.

## delete-fleet

Il seguente esempio di codice mostra come utilizzare `delete-fleet`.

### AWS CLI

Per eliminare un parco veicoli che non è più in uso

L'esempio seguente rimuove un parco istanze che è stato ridotto a zero istanze. Se la capacità del parco veicoli è maggiore di zero, la richiesta ha esito negativo e viene generato un errore HTTP 400.

```
aws gamelift delete-fleet \  
  --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Manage GameLift Fleets](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta [DeleteFleet AWS CLI](#) Command Reference.

## **delete-game-session-queue**

Il seguente esempio di codice mostra come utilizzare `delete-game-session-queue`.

### AWS CLI

Per eliminare una coda di sessione di gioco

L'`delete-game-session-queue` esempio seguente elimina una coda di sessione di gioco specificata.

```
aws gamelift delete-game-session-queue \  
  --name MegaFrogRace-NA
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteGameSessionQueue](#) in AWS CLI Command Reference.

## **describe-build**

Il seguente esempio di codice mostra come utilizzare `describe-build`.

### AWS CLI

Per ottenere informazioni sulla build di un gioco personalizzata

L'`describe-build` esempio seguente recupera le proprietà di una risorsa di build del server di gioco.

```
aws gamelift describe-build \  
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Caricare una build di server personalizzata GameLift](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta [DescribeBuild AWS CLI Command Reference](#).

## describe-ec2-instance-limits

Il seguente esempio di codice mostra come utilizzare `describe-ec2-instance-limits`.

### AWS CLI

Per recuperare i limiti di servizio per un tipo di EC2 istanza

L'esempio seguente visualizza il numero massimo di istanze consentite e le istanze correnti in uso per il tipo di EC2 istanza specificato nella regione corrente. Il risultato indica che vengono utilizzate solo cinque delle venti istanze consentite.

```
aws gamelift describe-ec2-instance-limits \
  --ec2-instance-type m5.Large
```

Output:

```
{
  "EC2InstanceLimits": [
    {
      "EC2InstanceType": "m5.large",
      "CurrentInstances": 5,
      "InstanceLimit": 20
    }
  ]
}
```

```
}

```

Per ulteriori informazioni, consulta [Choose Computing Resources](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta [DescribeEc2 InstanceLimits](#) in AWS CLI Command Reference.

## describe-fleet-attributes

Il seguente esempio di codice mostra come utilizzare `describe-fleet-attributes`.

### AWS CLI

Esempio 1: Per visualizzare gli attributi per un elenco di flotte

L'`describe-fleet-attributes` esempio seguente recupera gli attributi della flotta per due flotte specificate. Come illustrato, le flotte richieste vengono distribuite con la stessa build, una per le istanze On-Demand e una per le istanze Spot, con alcune differenze di configurazione minori.

```
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
```

Output:

```
{
  "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"
    }
}
]
}

```

## Esempio 2: per richiedere gli attributi per tutte le flotte

Quanto segue `describe-fleet-attributes` restituisce gli attributi della flotta per tutte le flotte con qualsiasi stato. Questo esempio illustra l'uso dei parametri di paginazione per restituire una flotta alla volta.

```

aws gamelift describe-fleet-attributes \
  --limit 1

```

## Output:

```
{
  "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":
  "eyJhd3NBWY2NvdW50SWQlOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjpw7InMiOiJidWlsZC01NWYxZTZmMS"
}
```

L'output include un NextToken valore che è possibile utilizzare quando si chiama il comando una seconda volta. Passate il valore al `--next-token` parametro per specificare dove prelevare l'output. Il comando seguente restituisce il secondo risultato nell'output.

```
aws gamelift describe-fleet-attributes \
  --limit 1 \
  --next-
token eyJhd3NBWY2NvdW50SWQlOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjpw7InMiOiJidWlsZC01NWYxZTZmMS
```

Ripetere l'operazione finché la risposta non include un NextToken valore.

Per ulteriori informazioni, consulta [Configurazione delle GameLift flotte](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta [DescribeFleetAttributes AWS CLI Command Reference](#).

## describe-fleet-capacity

Il seguente esempio di codice mostra come utilizzare `describe-fleet-capacity`.

### AWS CLI

Per visualizzare lo stato della capacità per un elenco di flotte

L'esempio seguente recupera la capacità attuale per due flotte specificate.

```
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
```

Output:

```
{
  "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
    }
}
]
}

```

Per ulteriori informazioni, consulta [GameLift Metrics for Fleets](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeFleetCapacity](#) Reference.

## describe-fleet-events

Il seguente esempio di codice mostra come utilizzare `describe-fleet-events`.

### AWS CLI

Per richiedere eventi per un periodo di tempo specificato

L'`describe-fleet-events` esempio seguente mostra i dettagli di tutti gli eventi relativi al parco veicoli che si sono verificati durante l'intervallo di tempo specificato.

```

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 \
  --limit 5

```

Output:

```

{
  "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",
    "EventTime": 1569319075.839,
    "PreSignedLogUrl": "https://gamelift-event-logs-prod-
us-west-2.s3.us-west-2.amazonaws.com/logs/fleet-83422059-8329-42a2-
a4d6-c4444386a6f8/events/2ecd0130-5986-44eb-99a7-62df27741084/
FLEET_VALIDATION_LAUNCH_PATH_NOT_FOUND.txt?X-Amz-Security-
Token=IQoJb3JpZ2luX2VjEB8aCXVzLXdlc3QtMiJHMEUCIHV5K%2FLPx8h310D
%2FAvx0%2FZxsDy5XA3cJ0wPdu3T0eBa%2FAiEA1yovokcZYy%2FV4CW6126aFyiSH0
%2Bxz%2FBMAhEHYHMqNcqkQMImp%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F
%2FARAAGgw3NDEwNjE0TlXNzEiDI8rsZtzLzlwEDQhXSrlAtl5Ae
%2Fgo6FCIzqXPbXfB0nSvFYqeDlriZarEpKqKrUt8mXQv9iqHResqCph9AKo49lwgSYTT2QoSxnRD7%2FUgv
%2BZm2pVuczvuKtUA0fcx6s0GxpjIAzdIE%2F5P%2FB7B9M%2BVZ
```

```

%2F9KF82hbJi0HTE6Y7BjKsEgFCvk4UXILhfjtan9iQ18%2F21ZTurAcJbm7Y5tuLF9SWSK3%2BEa7VX0cCK4D401sMj
%2FIaXoHkNvg0RVTa0hIqdvpaDQlsSBNdqTXbjHTu6fETE9Y9Ky%2BiJK5KiUG
%2F59GjCpDcvS1FqKeLUEmKT7wysGmvjMc2n%2Fr
%2F9VxQfte7w9srXwllLAQuwhiXAAyI5ICMz5JvzjzQwTqD4CHTVKUUDwL
%2BRZzbuuqkJOzbZm102CkRGp%2B74RTAzLbWptVqZTIIfzctiCTmWxb
%2FmKyELRYsVLrwNJ%2BGJ7%2BCrN0RC%2FjlgfLYIZyeAqjPgAu5HjgX
%2BM7jCo9M7wBTInAXK0FQuf9dvA84SuwX0JFp17LYGjrHMKv0qC3GfbTMrZ6kzeNV9awKCpXB2Gnx9z2KvI1JddqirW
%2F9C6%2B4jIZPME3jXmZcEHqqw5uvAVF7aeIavtUZU8pxpDIWT0YE4p3Kriy2AA7ziCRKtVfjv839InyLk8LUjsioWK
%2BYUq8%2FDT1Lxqj1S%2Fi04TI0Wo7ilAo%2FKKWWF4guuNDexj8E00ynSp1yImB
%2BZf2Fua3044W4eEXAMPLE33333&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20170621T231808Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20170621%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-
Signature=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
    }
  ],
  "NextToken":
    "eyJhd3NBZjY2NvdW50SWQiOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjpw7InMiOiJidWlsZC01NWYxZTZmMS"
}

```

Per ulteriori informazioni, consulta [Debug GameLift Fleet Issues](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta [DescribeFleetEvents AWS CLI Command Reference](#).

## describe-fleet-port-settings

Il seguente esempio di codice mostra come utilizzare `describe-fleet-port-settings`.

### AWS CLI

Per visualizzare le autorizzazioni di connessione in entrata per un parco veicoli

L'`describe-fleet-port-settings` seguente recupera le impostazioni di connessione per un parco veicoli specificato.

```

aws gamelift describe-fleet-port-settings \
  --fleet-id arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Output:

```

{
  "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"
}
]
```

Per ulteriori informazioni, consulta [Configurazione delle GameLift flotte](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta [DescribeFleetPortSettings AWS CLI Command Reference](#).

## describe-fleet-utilization

Il seguente esempio di codice mostra come utilizzare `describe-fleet-utilization`.

### AWS CLI

Esempio 1: per visualizzare i dati di utilizzo per un elenco di flotte

L'`describe-fleet-utilization` seguente recupera le informazioni sull'utilizzo corrente per una flotta specificata.

```
aws gamelift describe-fleet-utilization \
  --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{
  "FleetUtilization": [
    {
      "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "ActiveServerProcessCount": 100,
      "ActiveGameSessionCount": 62,
    }
  ]
}
```

```

    "CurrentPlayerSessionCount": 329,
    "MaximumPlayerSessionCount": 1000
  }
]
}

```

Esempio 2: per richiedere i dati di utilizzo per tutte le flotte

Quanto segue `describe-fleet-utilization` restituisce i dati di utilizzo della flotta per tutte le flotte con qualsiasi stato. Questo esempio utilizza i parametri di impaginazione per restituire i dati per due flotte alla volta.

```

aws gamelift describe-fleet-utilization \
  --limit 2

```

Output:

```

{
  "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": 329,
      "MaximumPlayerSessionCount": 1000
    }
  ],
  "NextToken":
  "eyJhd3NBWY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZEIkJj7InMi0iJidWlsZC01NWYxZTZmMS"
}

```

Chiamate il comando una seconda volta, passando il `NextToken` valore come argomento al `--next-token` parametro per vedere i due risultati successivi.

```

aws gamelift describe-fleet-utilization \

```

```
--limit 2 \
--next-
token eyJhd3NBY2NvdW50SWQiOmsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

Ripetete l'operazione finché la risposta non include più un NextToken valore nell'output.

Per ulteriori informazioni, consulta [GameLift Metrics for Fleets](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeFleetUtilization](#) Reference.

## describe-game-session-queues

Il seguente esempio di codice mostra come utilizzare `describe-game-session-queues`.

### AWS CLI

Per visualizzare le code delle sessioni di gioco

L'`describe-game-session-queues` esempio seguente recupera le proprietà per due code specificate.

```
aws gamelift describe-game-session-queues \
--names MegaFrogRace-NA MegaFrogRace-EU
```

Output:

```
{
  "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"
}
]
}

```

Per ulteriori informazioni, consulta [Using Multi-Region Queues](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeGameSessionQueues](#) Reference.

## describe-runtime-configuration

Il seguente esempio di codice mostra come utilizzare `describe-runtime-configuration`.

### AWS CLI

Per richiedere la configurazione di runtime per una flotta

L'`describe-runtime-configuration` esempio seguente recupera i dettagli sulla configurazione di runtime corrente per un parco veicoli specificato.

```
aws gamelift describe-runtime-configuration \
  --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Esegui più processi su una flotta](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta [DescribeRuntimeConfiguration AWS CLI](#) Command Reference.

## list-builds

Il seguente esempio di codice mostra come utilizzare `list-builds`.

### AWS CLI

Esempio 1: Per ottenere un elenco di build di gioco personalizzate

L'esempio seguente recupera le proprietà di tutte le build dei server di gioco nella regione corrente. La richiesta di esempio illustra come utilizzare i parametri di impaginazione

`Limit` e recuperare i `NextToken` risultati in set sequenziali. Il primo comando recupera le prime due build. Poiché sono disponibili più di due, la risposta include un `NextToken` per indicare che sono disponibili più risultati.

```
aws gamelift list-builds \  
--limit 2
```

Output:

```
{  
  "Builds": [  
    {  
      "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
      "CreationTime": 1495664528.723,  
      "Name": "My_Game_Server_Build_One",  
      "OperatingSystem": "WINDOWS_2012",  
      "SizeOnDisk": 8567781,  
      "Status": "READY",  
      "Version": "12345.678"  
    },  
    {  
      "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"  
    }  
  ],  
  "NextToken":  
    "eyJhd3NBZjY2NvdW50SWQ1Oj0sbnR5I6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjpw7InMi0iJidWlsZC01NWYxZTZmMS"  
}
```

È quindi possibile richiamare nuovamente il comando con il `--next-token` parametro seguente per visualizzare le due build successive.

```
aws gamelift list-builds \  
--next-token
```



```
--limit 2
--next-
token eyJhd3NBY2NvdW50SWQiOmsicyIGIjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

Ripeti finché la risposta non include un NextToken valore.

Esempio 2: per ottenere un elenco di build di giochi personalizzate in stato di errore

L'`list-builds` esempio seguente recupera le proprietà di tutte le build dei server di gioco nella regione corrente che hanno attualmente lo stato. FAILED

```
aws gamelift list-builds \
  --status FAILED
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, consulta AWS CLI Command [ListBuildsReference](#).

## list-fleets

Il seguente esempio di codice mostra come utilizzare `list-fleets`.

### AWS CLI

Esempio 1: Per ottenere un elenco di tutte le flotte in una regione

L'`list-fleets` seguente mostra la flotta IDs di tutte le flotte della regione corrente. Questo esempio utilizza i parametri di impaginazione per recuperare due flotte IDs alla volta. La risposta include un `next-token` attributo che indica che ci sono più risultati da recuperare.

```
aws gamelift list-fleets \  
  --limit 2
```

Output:

```
{  
  "FleetIds": [  
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"  
  ],  
  "NextToken":  
  "eyJhd3NBW50SWQ0I0nsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS"  
}
```

È possibile passare il `NextToken` valore della risposta precedente nel comando successivo, come illustrato di seguito, per ottenere i due risultati successivi.

```
aws gamelift list-fleets \  
  --limit 2 \  
  --next-  
token eyJhd3NBW50SWQ0I0nsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC00NDRLZj
```

Esempio 2: per ottenere un elenco di tutte le flotte in una regione con una build o uno script specifici

L'`list-builds` seguente recupera le flotte schierate con la build IDs di gioco specificata. Se lavori con Realtime Servers, puoi fornire un ID di script al posto di un ID di build. Poiché questo esempio non specifica il parametro `limit`, i risultati possono includere fino a 16 `fleetIDs`.

```
aws gamelift list-fleets \  
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{
```

```
"FleetIds": [  
  "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",  
  "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",  
  "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE44444"  
]  
}
```

- Per API i dettagli, vedere [ListFleets](#) in AWS CLI Command Reference.

## request-upload-credentials

Il seguente esempio di codice mostra come utilizzare `request-upload-credentials`.

### AWS CLI

Per aggiornare le credenziali di accesso per il caricamento di una build

L'`create-build` esempio seguente ottiene nuove credenziali di accesso valide per caricare un file di GameLift build in una posizione Amazon S3. Le credenziali hanno una durata limitata. L'ID di build viene ottenuto dalla risposta alla richiesta originale `CreateBuild`.

```
aws gamelift request-upload-credentials \  
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "StorageLocation": {  
    "Bucket": "gamelift-builds-us-west-2",  
    "Key": "123456789012/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
  },  
  "UploadCredentials": {  
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",  
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",  
    "SessionToken": "AgoGb3JpZ22luENZ...EXAMPLETOKEN=="  
  }  
}
```

Per ulteriori informazioni, consulta [Caricare una build di server personalizzata GameLift](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta [RequestUploadCredentials AWS CLI Command Reference](#).

## start-fleet-actions

Il seguente esempio di codice mostra come utilizzare `start-fleet-actions`.

### AWS CLI

Per riavviare l'attività di scalabilità automatica della flotta

L'esempio seguente riprende l'uso di tutte le politiche di scalabilità definite per il parco veicoli specificato, ma che sono state interrotte chiamando `stop-fleet-actions`. Dopo l'avvio, le politiche di scalabilità iniziano immediatamente a tracciare le rispettive metriche.

```
aws gamelift start-fleet-actions \  
  --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --actions AUTO_SCALING
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta AWS CLI Command [StartFleetActions Reference](#).

## stop-fleet-actions

Il seguente esempio di codice mostra come utilizzare `stop-fleet-actions`.

### AWS CLI

Per interrompere l'attività di scalabilità automatica di una flotta

L'esempio seguente interrompe l'uso di tutte le politiche di scalabilità definite per il parco veicoli specificato. Dopo la sospensione delle policy, la capacità della flotta rimane allo stesso numero di istanze attive, a meno che non venga regolata manualmente.

```
aws gamelift start-fleet-actions \  
  --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --actions AUTO_SCALING
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [StopFleetActions AWS CLI Command Reference](#).

## update-build

Il seguente esempio di codice mostra come utilizzare `update-build`.

### AWS CLI

Per aggiornare una build di gioco personalizzata

L'`update-build` seguente modifica le informazioni sul nome e sulla versione associate a una risorsa di build specificata. L'oggetto build restituito verifica che le modifiche siano state apportate correttamente.

```
aws gamelift update-build \  
  --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --name MegaFrogRaceServer.NA.east \  
  --build-version 12345.east
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Update Your Build Files](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta [UpdateBuild AWS CLI Command Reference](#).

## update-game-session-queue

Il seguente esempio di codice mostra come utilizzare `update-game-session-queue`.

## AWS CLI

Per aggiornare la configurazione della coda di una sessione di gioco

L'update-game-session-queueesempio seguente aggiunge una nuova destinazione e aggiorna i criteri di latenza dei giocatori per una coda di sessione di gioco esistente.

```
aws gamelift update-game-session-queue \
  --name MegaFrogRace-NA \
  --destinations file://destinations.json \
  --player-latency-policies file://latency-policies.json
```

Contenuto di 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"}
  ]
}
```

Contenuto di latency-policies.json.

```
{
  "PlayerLatencyPolicies": [
    {"MaximumIndividualPlayerLatencyMilliseconds": 200},
    {"MaximumIndividualPlayerLatencyMilliseconds": 150, "PolicyDurationSeconds":
120},
    {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":
120}
  ]
}
```

Output:

```
{
  "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}
    ]
}
}

```

Per ulteriori informazioni, consulta [Using Multi-Region Queues](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateGameSessionQueue](#) Reference.

## upload-build

Il seguente esempio di codice mostra come utilizzare `upload-build`.

### AWS CLI

Esempio 1: per caricare una build di un server di gioco Linux

L'`upload-build` esempio seguente carica i file di build del server di gioco Linux da una directory di file al GameLift servizio e crea una risorsa di compilazione.

```

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

```

## Output:

```
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
```

## Esempio 2: per caricare una build del server di gioco Windows

L'upload-build esempio seguente carica i file di build del server di gioco Windows da una directory al GameLift servizio e crea un record di build.

```
aws gamelift upload-build \
  --name MegaFrogRaceServer.NA \
  --build-version 2.0.1 \
  --build-root C:\MegaFrogRace_Server\release-na \
  --operating-system WINDOWS_2012
  --server-sdk-version 4.0.2
```

## Output:

```
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
```

Per ulteriori informazioni, consulta [Caricare una build di server personalizzata GameLift](#) nella Amazon GameLift Developer Guide.

- Per API i dettagli, consulta [UploadBuild AWS CLI](#) Command Reference.

## Esempi di Global Accelerator che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Global Accelerator. AWS Command Line Interface



Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **add-custom-routing-endpoints**

Il seguente esempio di codice mostra come utilizzare `add-custom-routing-endpoints`.

AWS CLI

Per aggiungere un endpoint di VPC sottorete a un gruppo di endpoint per un acceleratore di routing personalizzato

L'`add-custom-routing-endpoint` esempio seguente aggiunge un endpoint di VPC sottorete a un gruppo di endpoint per un acceleratore di routing personalizzato.

```
aws globalaccelerator add-custom-routing-endpoints \  
  --endpoint-group-  
  arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
  abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd \  
  --endpoint-configurations "EndpointId=subnet-1234567890abcdef0"
```

Output:

```
{  
  "EndpointDescriptions": [  
    {  
      "EndpointId": "subnet-1234567890abcdef0"  
    }  
  ],  
  
  "EndpointGroupArn": "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
  abcd-1234-abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd"
```

```
}
```

Per ulteriori informazioni, consulta gli [endpoint di VPC subnet per gli acceleratori di routing personalizzati in Global Accelerator nella Global Accelerator Developer Guide AWS](#).AWS

- Per i API dettagli, consulta Command Reference. [AddCustomRoutingEndpoints](#)AWS CLI

## **advertise-byoip-cidr**

Il seguente esempio di codice mostra come utilizzare `advertise-byoip-cidr`.

### AWS CLI

Per pubblicizzare un intervallo di indirizzi

L'`advertise-byoip-cidr` esempio seguente richiede AWS di pubblicizzare un intervallo di indirizzi che hai predisposto per l'uso con le tue risorse. AWS

```
aws globalaccelerator advertise-byoip-cidr \  
  --cidr 198.51.100.0/24
```

Output:

```
{  
  "ByoipCidr": {  
    "Cidr": "198.51.100.0/24",  
    "State": "PENDING_ADVERTISING"  
  }  
}
```

Per ulteriori informazioni, consulta [Bring Your Own IP Address in AWS Global Accelerator nella Global Accelerator Developer AWS Guide](#).

- Per API i dettagli, consulta AWS CLI Command [AdvertiseByoipCidr](#)Reference.

## **allow-custom-routing-traffic**

Il seguente esempio di codice mostra come utilizzare `allow-custom-routing-traffic`.

## AWS CLI

Per consentire il traffico verso destinazioni specifiche di EC2 istanze Amazon in una VPC sottorete per un acceleratore di routing personalizzato

L'`allow-custom-routing-traffic` seguente specifica che il traffico è consentito verso determinati indirizzi IP (destinazione) di EC2 istanze Amazon e che le porte di un endpoint di VPC sottorete in un acceleratore di routing personalizzato possono ricevere traffico.

```
aws globalaccelerator allow-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 "172.31.200.6" "172.31.200.7" \  
  --destination-ports 80 81
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta gli [endpoint di VPC sottorete per gli acceleratori di routing personalizzati in Global Accelerator nella Global Accelerator Developer Guide AWS](#).AWS

- Per i API dettagli, consulta Command Reference. [AllowCustomRoutingTraffic](#)AWS CLI

## `create-accelerator`

Il seguente esempio di codice mostra come utilizzare `create-accelerator`.

## AWS CLI

Per creare un acceleratore

L'`create-accelerator` seguente crea un acceleratore con due tag con due indirizzi IP BYOIP statici. È necessario specificare la US-West-2 (Oregon) regione per creare o aggiornare un acceleratore.

```
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
```

Output:

```
{
  "Accelerator": {
    "AcceleratorArn":
"arn:aws:globalaccelerator::012345678901: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": 1542394847.0,
    "LastModifiedTime": 1542394847.0
  }
}
```

Per ulteriori informazioni, consulta [Accelerators in AWS Global Accelerator nella Global Accelerator](#) Developer AWS Guide.

- Per API i dettagli, consulta Command [CreateAccelerator](#) Reference AWS CLI .

## create-custom-routing-accelerator

Il seguente esempio di codice mostra come utilizzare `create-custom-routing-accelerator`.

### AWS CLI

Per creare un acceleratore di routing personalizzato

L'esempio seguente crea un acceleratore di routing personalizzato con i tag e. Name Project

```
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
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta la sezione [Acceleratori di routing personalizzati in Global Accelerator nella AWS Global Accelerator Developer Guide AWS](#).

- Per i API dettagli, consulta [CreateCustomRoutingAccelerator](#) Command Reference.AWS CLI

## create-custom-routing-endpoint-group

Il seguente esempio di codice mostra come utilizzare `create-custom-routing-endpoint-group`.

AWS CLI

Per creare un gruppo di endpoint per un acceleratore di routing personalizzato

L'create-custom-routing-endpoint-groupesempio seguente crea un gruppo di endpoint per un acceleratore di routing personalizzato.

```
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"
```

Output:

```
{
  "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": []
  }
}
```

Per ulteriori informazioni, consulta [Endpoint groups for custom routing accelerators in AWS Global Accelerator nella Global Accelerator Developer Guide](#).AWS

- Per i API dettagli, consulta Command Reference. [CreateCustomRoutingEndpointGroup](#)AWS CLI

## create-custom-routing-listener

Il seguente esempio di codice mostra come utilizzarecreate-custom-routing-listener.

## AWS CLI

Per creare un listener per un acceleratore di routing personalizzato

L'`create-custom-routing-listener` seguente crea un listener con un intervallo di porte compreso tra 5000 e 10000 per un acceleratore di routing personalizzato.

```
aws globalaccelerator create-custom-routing-listener \  
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \  
  --port-ranges FromPort=5000,ToPort=10000
```

Output:

```
{  
  "Listener": {  
    "PortRange": [  
      "FromPort": 5000,  
      "ToPort": 10000  
    ],  
    "ListenerArn":  
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/0123vxyz"  
  }  
}
```

Per ulteriori informazioni, consulta [Listeners for custom routing accelerators in AWS Global Accelerator nella Global Accelerator Developer Guide](#).AWS

- Per i API dettagli, consulta Command Reference. [CreateCustomRoutingListener](#) AWS CLI

## `create-endpoint-group`

Il seguente esempio di codice mostra come utilizzare `create-endpoint-group`.

## AWS CLI

Per creare un gruppo di endpoint

L'`create-endpoint-group` seguente crea un gruppo di endpoint con un endpoint.

```
aws globalaccelerator create-endpoint-group \  
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \  
  --endpoint-group-name my-endpoint-group \  
  --endpoints arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh:listener/0123vxyz
```

```

--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

```

Output:

```

{
  "EndpointGroup": {
    "TrafficDialPercentage": 100.0,
    "EndpointDescriptions": [
      {
        "Weight": 128,
        "EndpointId": "i-1234567890abcdef0"
      }
    ],
    "EndpointGroupArn":
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/098765zyxwvu",
    "EndpointGroupRegion": "us-east-1"
  }
}

```

Per ulteriori informazioni, consulta i [gruppi di endpoint in Global Accelerator nella AWS Global Accelerator Developer Guide AWS](#) .

- Per API i dettagli, consulta Command [CreateEndpointGroup](#) Reference AWS CLI .

## create-listener

Il seguente esempio di codice mostra come utilizzare `create-listener`.

### AWS CLI

Per creare un ascoltatore

L'`create-listener` esempio seguente crea un listener con due porte.

```

aws globalaccelerator create-listener \
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh \
  --port-ranges FromPort=80,ToPort=80 FromPort=81,ToPort=81 \

```



```
--protocol TCP
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Listeners in AWS Global Accelerator nella Global Accelerator Developer AWS Guide](#).

- Per API i dettagli, consulta Command [CreateListener](#)Reference AWS CLI .

## deny-custom-routing-traffic

Il seguente esempio di codice mostra come utilizzare `deny-custom-routing-traffic`.

AWS CLI

Per specificare un indirizzo di destinazione che non può ricevere traffico in un acceleratore di routing personalizzato

L'`deny-custom-routing-traffic`esempio seguente specifica l'indirizzo o gli indirizzi di destinazione in un endpoint di sottorete che non possono ricevere traffico per un acceleratore di routing personalizzato. Per specificare più di un indirizzo di destinazione, separa gli indirizzi con uno spazio. Nessuna risposta per una `deny-custom-routing-traffic` chiamata andata a buon fine.

```
aws globalaccelerator deny-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 "198.51.100.52"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta gli [endpoint di VPC sottorete per gli acceleratori di routing personalizzati in Global Accelerator nella AWS Global Accelerator Developer Guide](#).AWS

- Per i API dettagli, consulta Command Reference. [DenyCustomRoutingTraffic](#)AWS CLI

## deprovision-byoip-cidr

Il seguente esempio di codice mostra come utilizzare `deprovision-byoip-cidr`.

### AWS CLI

Per rimuovere il provisioning di un intervallo di indirizzi

L'`deprovision-byoip-cidr` seguente rilascia l'intervallo di indirizzi specificato che avete predisposto per l'utilizzo con le vostre AWS risorse.

```
aws globalaccelerator deprovision-byoip-cidr \  
  --cidr "198.51.100.0/24"
```

Output:

```
{  
  "ByoipCidr": {  
    "Cidr": "198.51.100.0/24",  
    "State": "PENDING_DEPROVISIONING"  
  }  
}
```

Per ulteriori informazioni, consulta [Bring your own IP address in AWS Global Accelerator nella Global Accelerator AWS Developer Guide](#).

- Per API i dettagli, consulta AWS CLI Command [DeprovisionByoipCidr](#)Reference.

## describe-accelerator-attributes

Il seguente esempio di codice mostra come utilizzare `describe-accelerator-attributes`.

### AWS CLI

Per descrivere gli attributi di un acceleratore

L'esempio seguente recupera i dettagli degli attributi per un acceleratore.

```
aws globalaccelerator describe-accelerator-attributes \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
  abcd-1234-abcd-1234abcdefgh
```

Output:

```
{
  "AcceleratorAttributes": {
    "FlowLogsEnabled": true
    "FlowLogsS3Bucket": flowlogs-abc
    "FlowLogsS3Prefix": bucketprefix-abc
  }
}
```

Per ulteriori informazioni, consulta [Accelerators in AWS Global Accelerator nella Global Accelerator Developer Guide AWS](#).

- Per API i dettagli, consulta Command [DescribeAcceleratorAttributes](#) Reference AWS CLI .

## describe-accelerator

Il seguente esempio di codice mostra come utilizzare `describe-accelerator`.

### AWS CLI

Per descrivere un acceleratore

L'esempio seguente recupera i dettagli sull'acceleratore specificato.

```
aws globalaccelerator describe-accelerator \
```

```
--accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Accelerators in AWS Global Accelerator nella Global Accelerator Developer Guide AWS](#).

- Per API i dettagli, consulta Command [DescribeAccelerator](#) Reference AWS CLI.

## **describe-custom-routing-accelerator-attributes**

Il seguente esempio di codice mostra come utilizzare `describe-custom-routing-accelerator-attributes`.

AWS CLI

Per descrivere gli attributi di un acceleratore di routing personalizzato

L'`describe-custom-routing-accelerator-attributes` seguente descrive gli attributi di un acceleratore di routing personalizzato.

```
aws globalaccelerator describe-custom-routing-accelerator-attributes \  
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh
```

Output:

```
{  
  "AcceleratorAttributes": {  
    "FlowLogsEnabled": false  
  }  
}
```

Per ulteriori informazioni, consulta la sezione [Acceleratori di routing personalizzati in Global Accelerator nella AWS Global Accelerator Developer Guide AWS](#) .

- Per i API dettagli, consulta [DescribeCustomRoutingAcceleratorAttributesCommand Reference.AWS CLI](#)

## `describe-custom-routing-accelerator`

Il seguente esempio di codice mostra come utilizzare `describe-custom-routing-accelerator`.

AWS CLI

Per descrivere un acceleratore di routing personalizzato

L'`describe-custom-routing-accelerator` seguente recupera i dettagli sull'acceleratore di routing personalizzato specificato.

```
aws globalaccelerator describe-custom-routing-accelerator \  
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh
```

Output:

```
{  
  "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
    }
  }
```

Per ulteriori informazioni, consulta la sezione [Acceleratori di routing personalizzati in Global Accelerator nella AWS Global Accelerator Developer Guide](#).AWS

- Per i API dettagli, consulta [DescribeCustomRoutingAccelerator](#) Command Reference.AWS CLI

## **describe-custom-routing-endpoint-group**

Il seguente esempio di codice mostra come utilizzare `describe-custom-routing-endpoint-group`.

AWS CLI

Per descrivere un gruppo di endpoint per un acceleratore di routing personalizzato

L'esempio seguente descrive un gruppo di endpoint per un acceleratore di routing personalizzato.

```
aws globalaccelerator describe-custom-routing-endpoint-group \
```

**--endpoint-group-**

**arn *arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefg*h/listener/6789vxyz/endpoint-group/ab8888example**

Output:

```
{
  "EndpointGroup": {
    "EndpointGroupArn":
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
      abcd-1234abcdefg
```

h/listener/6789vxyz/endpoint-group/ab8888example",
 "EndpointGroupRegion": "us-east-2",
 "DestinationDescriptions": [
 {
 "FromPort": 5000,
 "ToPort": 10000,
 "Protocols": [
 "UDP"
 ]
 }
 ],
 "EndpointDescriptions": [
 {
 "EndpointId": "subnet-1234567890abcdef0"
 }
 ]
 }
}

Per ulteriori informazioni, consulta [Endpoint groups for custom routing accelerators in AWS Global Accelerator nella Global Accelerator Developer Guide](#).AWS

- Per i API dettagli, consulta Command Reference. [DescribeCustomRoutingEndpointGroup](#)AWS CLI

## **describe-custom-routing-listener**

Il seguente esempio di codice mostra come utilizzare `describe-custom-routing-listener`.

AWS CLI

Per descrivere un listener per un acceleratore di routing personalizzato

L'`describe-custom-routing-listener` seguente descrive un listener per un acceleratore di routing personalizzato.

```
aws globalaccelerator describe-custom-routing-listener \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

Output:

```
{
  "Listener": {
    "PortRanges": [
      "FromPort": 5000,
      "ToPort": 10000
    ],
    "ListenerArn":
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234"
  }
}
```

Per ulteriori informazioni, consulta [Listeners for custom routing accelerators in AWS Global Accelerator nella Global Accelerator Developer Guide](#).AWS

- Per i API dettagli, consulta Command Reference. [DescribeCustomRoutingListener](#)AWS CLI

## `describe-endpoint-group`

Il seguente esempio di codice mostra come utilizzare `describe-endpoint-group`.

AWS CLI

Per descrivere un gruppo di endpoint

L'`describe-endpoint-group` seguente recupera i dettagli su un gruppo di endpoint con i seguenti endpoint: un'EC2istanza AmazonALB, an e an. 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/
ab8888example
```



**Output:**

```
{
  "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-1234abcdefggh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-
group/4321abcd-abcd-4321-abcd-4321abcdefg",
    "EndpointGroupRegion": "us-east-1"
  }
}
```

Per ulteriori informazioni, consulta i [gruppi di endpoint in Global Accelerator nella AWS Global Accelerator Developer Guide AWS](#) .

- Per API i dettagli, consulta Command [DescribeEndpointGroup](#)Reference AWS CLI .

**describe-listener**

Il seguente esempio di codice mostra come utilizzare `describe-listener`.

**AWS CLI**

Per descrivere un ascoltatore

L'`describe-listener`esempio seguente descrive un ascoltatore.

```
aws globalaccelerator describe-listener \  
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

Output:

```
{  
  "Listener": {  
    "ListenerArn":  
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/abcdef1234",  
    "PortRanges": [  
      {  
        "FromPort": 80,  
        "ToPort": 80  
      }  
    ],  
    "Protocol": "TCP",  
    "ClientAffinity": "NONE"  
  }  
}
```

Per ulteriori informazioni, consulta [Listeners in AWS Global Accelerator nella Global Accelerator Developer AWS Guide](#).

- Per API i dettagli, consulta Command [DescribeListener](#)Reference AWS CLI .

## list-accelerators

Il seguente esempio di codice mostra come utilizzare `list-accelerators`.

AWS CLI

Per elencare i tuoi acceleratori

L'`list-accelerators`esempio seguente elenca gli acceleratori presenti nel tuo AWS account. Questo account ha due acceleratori.

```
aws globalaccelerator list-accelerators
```

Output:



```
    },  
  ]  
}
```

Per ulteriori informazioni, consulta [Accelerators in AWS Global Accelerator nella Global Accelerator Developer AWS Guide](#).

- Per API i dettagli, consulta Command [ListAccelerators](#)Reference AWS CLI .

## **list-byoip-cidr**

Il seguente esempio di codice mostra come utilizzare `list-byoip-cidr`.

### AWS CLI

Per elencare gli intervalli di indirizzi

L'`list-byoip-cidr` esempio seguente elenca gli intervalli di indirizzi Bring your own IP address (BYOIP) che hai predisposto per l'uso con Global Accelerator.

```
aws globalaccelerator list-byoip-cidrs
```

Output:

```
{  
  "ByoipCidrs": [  
    {  
      "Cidr": "198.51.100.0/24",  
      "State": "READY"  
    },  
    {  
      "Cidr": "203.0.113.25/24",  
      "State": "READY"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Bring your own IP address in AWS Global Accelerator nella Global Accelerator Developer AWS Guide](#).

- Per API i dettagli, consulta AWS CLI Command [ListByoipCidr](#)Reference.





```

    {
      "EndpointGroupArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234/endpoint-group/ab88888example",
      "EndpointGroupRegion": "eu-central-1",
      "DestinationDescriptions": [
        {
          "FromPort": 80,
          "ToPort": 80,
          "Protocols": [
            "TCP",
            "UDP"
          ]
        }
      ]
      "EndpointDescriptions": [
        {
          "EndpointId": "subnet-abcd123example"
        }
      ]
    }
  ]
}

```

Per ulteriori informazioni, consulta [Endpoint groups for custom routing accelerators in AWS Global Accelerator nella Global Accelerator Developer Guide](#).AWS

- Per i API dettagli, consulta Command Reference. [ListCustomRoutingEndpointGroups](#)AWS CLI

## list-custom-routing-listeners

Il seguente esempio di codice mostra come utilizzare `list-custom-routing-listeners`.

### AWS CLI

Per elencare i listener per gli acceleratori di routing personalizzati

L'`list-custom-routing-listeners`esempio seguente elenca i listener per un acceleratore di routing personalizzato.

```

aws globalaccelerator list-custom-routing-listeners \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh

```

**Output:**

```
{
  "Listeners": [
    {
      "ListenerArn":
      "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
      abcd-1234abcdefggh/listener/abcdef1234",
      "PortRanges": [
        {
          "FromPort": 5000,
          "ToPort": 10000
        }
      ],
      "Protocol": "TCP"
    }
  ]
}
```

Per ulteriori informazioni, consultate [Listeners for custom routing accelerators in AWS Global Accelerator nella Global Accelerator Developer Guide](#).AWS

- Per i API dettagli, consulta Command Reference. [ListCustomRoutingListeners](#)AWS CLI

**list-custom-routing-port-mappings-by-destination**

Il seguente esempio di codice mostra come utilizzare `list-custom-routing-port-mappings-by-destination`.

**AWS CLI**

Per elencare le mappature delle porte per una destinazione specifica dell'acceleratore di routing personalizzato

L'`list-custom-routing-port-mappings-by-destination`esempio seguente fornisce le mappature delle porte per un EC2 server di destinazione specifico (all'indirizzo di destinazione) per un acceleratore di routing personalizzato.

```
aws globalaccelerator list-custom-routing-port-mappings-by-destination \
  --endpoint-id subnet-abcd123example \
  --destination-address 198.51.100.52
```



**Output:**

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Come funzionano gli acceleratori di routing personalizzati in Global Accelerator nella AWS Global Accelerator Developer Guide](#).AWS

- Per i API dettagli, consulta [ListCustomRoutingPortMappingsByDestination](#) Command Reference.AWS CLI

**list-custom-routing-port-mappings**

Il seguente esempio di codice mostra come utilizzare `list-custom-routing-port-mappings`.

## AWS CLI

Per elencare le mappature delle porte in un acceleratore di routing personalizzato

L'`list-custom-routing-port-mappings` seguente fornisce un elenco parziale delle mappature delle porte in un acceleratore di routing personalizzato.

```
aws globalaccelerator list-custom-routing-port-mappings \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh
```

Output:

```
{
  "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"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Come funzionano gli acceleratori di routing personalizzati in Global Accelerator nella AWS Global Accelerator Developer Guide](#).AWS

- Per i API dettagli, consulta [ListCustomRoutingPortMappings](#) Command Reference.AWS CLI

## list-endpoint-groups

Il seguente esempio di codice mostra come utilizzare `list-endpoint-groups`.

AWS CLI

Per elencare i gruppi di endpoint

L'`list-endpoint-groups`esempio seguente elenca i gruppi di endpoint per un listener. Questo listener ha due gruppi di endpoint.

```

aws globalaccelerator --region us-west-2 list-endpoint-groups \
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234

```

Output:

```

{
  "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
    }
  ]
}

```

Per ulteriori informazioni, consulta [Endpoint Groups in AWS Global Accelerator nella Global Accelerator Developer Guide AWS](#) .

- Per API i dettagli, consulta Command [ListEndpointGroups](#) Reference AWS CLI .

## list-listeners

Il seguente esempio di codice mostra come utilizzare `list-listeners`.

### AWS CLI

Per elencare gli ascoltatori

L'`list-listeners` esempio seguente elenca gli ascoltatori di un acceleratore.

```

aws globalaccelerator list-listeners \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh

```

Output:

```

{
  "Listeners": [
    {
      "ListenerArn":
"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234",

```

```

        "PortRanges": [
            {
                "FromPort": 80,
                "ToPort": 80
            }
        ],
        "Protocol": "TCP",
        "ClientAffinity": "NONE"
    }
]
}

```

Per ulteriori informazioni, consultate [Listeners in AWS Global Accelerator nella Global Accelerator Developer Guide AWS](#) .

- Per API i dettagli, consulta Command [ListListeners](#)Reference AWS CLI .

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di un acceleratore

L'`list-tags-for-resource`esempio seguente elenca i tag per un acceleratore specifico.

```

aws globalaccelerator list-tags-for-resource \
  --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh

```

Output:

```

{
  "Tags": [
    {
      "Key": "Project",
      "Value": "A123456"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Tagging in AWS Global Accelerator nella Global Accelerator Developer AWS Guide](#).

- Per API i dettagli, consulta Command [ListTagsForResource](#)Reference AWS CLI .

## provision-byoip-cidr

Il seguente esempio di codice mostra come utilizzare `provision-byoip-cidr`.

### AWS CLI

Per fornire un intervallo di indirizzi

L'`provision-byoip-cidr`esempio seguente fornisce l'intervallo di indirizzi specificato da utilizzare con le AWS risorse.

```
aws globalaccelerator provision-byoip-cidr \  
  --cidr 192.0.2.250/24 \  
  --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

Output:

```
{  
  "ByoipCidr": {  
    "Cidr": "192.0.2.250/24",  
    "State": "PENDING_PROVISIONING"  
  }  
}
```

Per ulteriori informazioni, consulta [Bring your own IP address in AWS Global Accelerator nella Global Accelerator Developer Guide](#).AWS

- Per API i dettagli, consulta AWS CLI Command [ProvisionByoipCidr](#)Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare un acceleratore

L'`tag-resource` seguente aggiunge i tag `Name` e `Project` a un acceleratore, insieme ai valori corrispondenti per ciascuno di essi.

```
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"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging in AWS Global Accelerator nella Global Accelerator Developer AWS Guide](#).

- Per API i dettagli, consulta Command [TagResource](#) Reference AWS CLI .

## **untag-resource**

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da un acceleratore

L'`untag-resource` seguente rimuove i tag `Name` e `Project` da un acceleratore.

```
aws globalaccelerator untag-resource \  
  --resource-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \  
  --tag-keys Key="Name" Key="Project"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging in AWS Global Accelerator nella Global Accelerator Developer AWS Guide](#).

- Per API i dettagli, consulta Command [UntagResource](#) Reference AWS CLI .

## **update-accelerator-attributes**

Il seguente esempio di codice mostra come utilizzare `update-accelerator-attributes`.

## AWS CLI

Per aggiornare gli attributi di un acceleratore

L'update-accelerator-attributesesempio seguente aggiorna un acceleratore per abilitare i registri di flusso. È necessario specificare la US-West-2 (Oregon) regione per creare o aggiornare gli attributi dell'acceleratore.

```
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
```

Output:

```
{  
  "AcceleratorAttributes": {  
    "FlowLogsEnabled": true  
    "FlowLogsS3Bucket": flowlogs-abc  
    "FlowLogsS3Prefix": bucketprefix-abc  
  }  
}
```

Per ulteriori informazioni, consulta [Accelerators in AWS Global Accelerator nella Global Accelerator Developer AWS Guide](#).

- Per API i dettagli, consulta Command [UpdateAcceleratorAttributes](#)Reference AWS CLI .

## update-accelerator

Il seguente esempio di codice mostra come utilizzareupdate-accelerator.

## AWS CLI

Per aggiornare un acceleratore

L'update-acceleratoroesempio seguente modifica un acceleratore per cambiarne il nome. ExampleAcceleratorNew È necessario specificare la US-West-2 (Oregon) regione per creare o aggiornare gli acceleratori.



```
aws globalaccelerator update-accelerator \  
  --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \  
  --name ExampleAcceleratorNew
```

Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, consulta [Accelerators in AWS Global Accelerator nella Global Accelerator Developer AWS Guide](#).

- Per API i dettagli, consulta Command [UpdateAccelerator](#) Reference AWS CLI .

## update-custom-routing-accelerator-attributes

Il seguente esempio di codice mostra come utilizzare `update-custom-routing-accelerator-attributes`.

## AWS CLI

Per aggiornare gli attributi di un acceleratore di routing personalizzato

L'update-custom-routing-accelerator-attributesesempio seguente aggiorna un acceleratore di routing personalizzato per abilitare i log di flusso.

```
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
```

Output:

```
{  
  "AcceleratorAttributes": {  
    "FlowLogsEnabled": true  
    "FlowLogsS3Bucket": flowlogs-abc  
    "FlowLogsS3Prefix": bucketprefix-abc  
  }  
}
```

Per ulteriori informazioni, consulta [Acceleratori di routing personalizzati in Global Accelerator nella AWS Global Accelerator Developer Guide](#).AWS

- Per i API dettagli, consulta [UpdateCustomRoutingAcceleratorAttributesCommand Reference](#).AWS CLI

## update-custom-routing-accelerator

Il seguente esempio di codice mostra come utilizzareupdate-custom-routing-accelerator.

## AWS CLI

Per aggiornare un acceleratore di routing personalizzato

L'update-custom-routing-acceleratoresempio seguente modifica un acceleratore di routing personalizzato per cambiare il nome dell'acceleratore.

```
aws globalaccelerator --region us-west-2 update-custom-routing-accelerator \  
  --accelerator-name new-accelerator-name
```

```
--accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh \  
--name ExampleCustomRoutingAcceleratorNew
```

Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, consulta la sezione [Acceleratori di routing personalizzati in Global Accelerator nella AWS Global Accelerator Developer Guide](#).AWS

- Per i API dettagli, consulta [UpdateCustomRoutingAccelerator](#) Command Reference.AWS CLI

## update-custom-routing-listener

Il seguente esempio di codice mostra come utilizzare `update-custom-routing-listener`.

AWS CLI

Per aggiornare un listener per un acceleratore di routing personalizzato

L'update-custom-routing-listeneresempio seguente aggiorna un listener per modificare l'intervallo di porte.

```
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
```

Output:

```
{
  "Listener": {
    "ListenerArn":
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz
    "PortRanges": [
      {
        "FromPort": 10000,
        "ToPort": 20000
      }
    ],
    "Protocol": "TCP"
  }
}
```

Per ulteriori informazioni, consulta [Listeners for custom routing accelerators in AWS Global Accelerator nella Global Accelerator Developer Guide](#).AWS

- Per i API dettagli, consulta Command Reference. [UpdateCustomRoutingListener](#)AWS CLI

## update-endpoint-group

Il seguente esempio di codice mostra come utilizzareupdate-endpoint-group.

AWS CLI

Per aggiornare un gruppo di endpoint

L'update-endpoint-groupesempio seguente aggiunge tre endpoint a un gruppo di endpoint: un indirizzo IP elasticoALB, un e un. NLB

```
aws globalaccelerator update-endpoint-group \
```

```

--endpoint-group-
arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefg/Listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-group/
ab8888example \
  --endpoint-configurations \
    EndpointId=eipalloc-eip01234567890abc,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,Weight=128

```

Output:

```

{
  "EndpointGroup": {
    "TrafficDialPercentage": 100,
    "EndpointDescriptions": [
      {
        "Weight": 128,
        "EndpointId": "eip01234567890abc"
      },
      {
        "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::123456789012:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefg/Listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-
group/4321abcd-abcd-4321-abcd-4321abcdefg",
    "EndpointGroupRegion": "us-east-1"
  }
}

```

Per ulteriori informazioni, consulta i [gruppi di endpoint in Global Accelerator nella AWS Global Accelerator Developer Guide AWS](#).

- Per API i dettagli, consulta Command [UpdateEndpointGroup](#) Reference AWS CLI .

## update-listener

Il seguente esempio di codice mostra come utilizzare `update-listener`.

### AWS CLI

Per aggiornare un ascoltatore

L'`update-listener` esempio seguente aggiorna un listener per modificare la porta su 100.

```
aws globalaccelerator update-listener \  
  --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-  
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \  
  --port-ranges FromPort=100,ToPort=100
```

Output:

```
{  
  "Listener": {  
    "ListenerArn":  
    "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-  
abcd-1234abcdefgh/listener/0123vxyz"  
    "PortRanges": [  
      {  
        "FromPort": 100,  
        "ToPort": 100  
      }  
    ],  
    "Protocol": "TCP",  
    "ClientAffinity": "NONE"  
  }  
}
```

Per ulteriori informazioni, consulta [Listeners in AWS Global Accelerator nella Global Accelerator Developer AWS Guide](#).

- Per API i dettagli, consulta Command [UpdateListener](#) Reference AWS CLI .

## withdraw-byoip-cidr

Il seguente esempio di codice mostra come utilizzare `withdraw-byoip-cidr`.

### AWS CLI

Per ritirare un intervallo di indirizzi

L'`withdraw-byoip-cidr` seguente rimuove da AWS Global Accelerator un intervallo di indirizzi che avevi precedentemente pubblicizzato per l'utilizzo con le tue risorse. AWS

```
aws globalaccelerator withdraw-byoip-cidr \  
  --cidr 192.0.2.250/24
```

Output:

```
{  
  "ByoipCidr": {  
    "Cidr": "192.0.2.250/24",  
    "State": "PENDING_WITHDRAWING"  
  }  
}
```

Per ulteriori informazioni, consulta [Bring your own IP address in AWS Global Accelerator nella Global Accelerator Developer Guide AWS](#).

- Per API i dettagli, consulta AWS CLI Command [WithdrawByoipCidrReference](#).

## AWS Glue esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Glue.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **batch-stop-job-run**

Il seguente esempio di codice mostra come utilizzare `batch-stop-job-run`.

#### AWS CLI

Per interrompere l'esecuzione dei job

L'`batch-stop-job-run` seguente interrompe l'esecuzione di un processo.

```
aws glue batch-stop-job-run \  
  --job-name "my-testing-job" \  
  --job-run-id jr_852f1de1f29fb62e0ba4166c33970803935d87f14f96cfdee5089d5274a61d3f
```

Output:

```
{  
  "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  
  }  
}
```



Per ulteriori informazioni, consulta [Esecuzioni di processi](#) nella Guida per gli sviluppatori di AWS Glue.

- Per API i dettagli, vedere [BatchStopJobRunin](#) in AWS CLI Command Reference.

## create-connection

Il seguente esempio di codice mostra come utilizzare `create-connection`.

### AWS CLI

Per creare una connessione per gli archivi dati di AWS Glue

L'`create-connection`esempio seguente crea una connessione nel AWS Glue Data Catalog che fornisce informazioni di connessione per un data store Kafka.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Defining Connections in the AWS Glue Data Catalog](#) nella AWS Glue Developer Guide.

- Per API i dettagli, vedere [CreateConnection](#) in AWS CLI Command Reference.

## create-database

Il seguente esempio di codice mostra come utilizzare `create-database`.

### AWS CLI

Per creare un database:

L'esempio seguente crea un database nel AWS Glue Data Catalog.

```
aws glue create-database \  
  --database-input "{\"Name\":\"tempdb\"}" \  
  --profile my_profile \  
  --endpoint https://glue.us-east-1.amazonaws.com
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Definizione di un database nel catalogo dati](#) nella Guida per gli sviluppatori di AWS Glue.

- Per API i dettagli, vedere [CreateDatabase](#) in AWS CLI Command Reference.

## create-job

Il seguente esempio di codice mostra come utilizzare `create-job`.

### AWS CLI

Per creare un processo di trasformazione dei dati

L'esempio `create-job` seguente crea un processo di streaming che esegue uno script archiviato in S3.

```
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

```

Contenuto di 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()
}
}

```

Output:

```

{
  "Name": "my-testing-job"
}

```

Per ulteriori informazioni, consulta [Authoring Jobs in AWS Glue nella Glue Developer Guide](#).AWS

- Per API i dettagli, consulta [CreateJob AWS CLI Command Reference](#).

## create-table

Il seguente esempio di codice mostra come utilizzare `create-table`.

## AWS CLI

Esempio 1: creare una tabella per un flusso di dati Kinesis

L'`create-table`esempio seguente crea una tabella nel AWS Glue Data Catalog che descrive un flusso di dati Kinesis.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Definizione delle tabelle nel AWS Glue Data Catalog](#) nella AWS Glue Developer Guide.

Esempio 2: Per creare una tabella per un data store Kafka

L'`create-table`esempio seguente crea una tabella nel AWS Glue Data Catalog che descrive un data store Kafka.

```
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":"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
```

```

        {"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

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Definizione delle tabelle nel AWS Glue Data Catalog](#) nella AWS Glue Developer Guide.

Esempio 3: Per creare una tabella per un data AWS store S3

L'create-tableesempio seguente crea una tabella nel AWS Glue Data Catalog che descrive un data store AWS Simple Storage Service (AWS S3).

```

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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Definizione delle tabelle nel AWS Glue Data Catalog](#) nella AWS Glue Developer Guide.

- Per API i dettagli, vedere [CreateTable](#) in AWS CLI Command Reference.

## delete-job

Il seguente esempio di codice mostra come utilizzare `delete-job`.

### AWS CLI

Per eliminare un processo

L'esempio `delete-job` seguente elimina un processo non più necessario.

```
aws glue delete-job \  
--job-name my-testing-job
```

Output:

```
{  
  "JobName": "my-testing-job"  
}
```

Per ulteriori informazioni, consulta [Working with Jobs on the AWS Glue Console](#) nella AWS Glue Developer Guide.

- Per API i dettagli, consulta [DeleteJob AWS CLI](#) Command Reference.

## get-databases

Il seguente esempio di codice mostra come utilizzare `get-databases`.

### AWS CLI

Per elencare le definizioni di alcuni o tutti i database del AWS Glue Data Catalog

L'esempio `get-databases` seguente restituisce informazioni sui database del Catalogo dati.

```
aws glue get-databases
```

Output:

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Definizione di un database nel catalogo dati](#) nella Guida per gli sviluppatori di AWS Glue.

- Per API i dettagli, consultate [GetDatabases AWS CLI](#) Command Reference.

## get-job-run

Il seguente esempio di codice mostra come utilizzare `get-job-run`.

### AWS CLI

Per ottenere informazioni relative all'esecuzione di un processo

L'esempio `get-job-run` seguente recupera le informazioni relative all'esecuzione di un processo.

```
aws glue get-job-run \  
  --job-name "Combine legislators data" \  
  --run-id jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Esecuzioni di processi](#) nella Guida per gli sviluppatori di AWS Glue.

- Per API i dettagli, vedere [GetJobRun](#) in AWS CLI Command Reference.

## get-job-runs

Il seguente esempio di codice mostra come utilizzare `get-job-runs`.

### AWS CLI

Per ottenere informazioni su tutte le esecuzioni di processo per un determinato processo

L'esempio `get-job-runs` seguente recupera informazioni sulle esecuzioni di processo per un determinato processo.

```
aws glue get-job-runs \  
  --job-name "my-testing-job"
```

Output:

```
{  
  "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
+PpC6Gp0RahyF2fp9f1B5SSb2bTGPnUSPVizLXR11PN3QZ1db+v1o9qRVktNYbW8=)",
    "PredecessorRuns": [],
    "AllocatedCapacity": 10,
    "ExecutionTime": 113,
    "Timeout": 2880,
    "MaxCapacity": 10.0,
    "WorkerType": "G.1X",
    "NumberOfWorkers": 10,
    "LogGroupName": "/aws-glue/jobs",
    "GlueVersion": "2.0"
  }
]
}

```

Per ulteriori informazioni, consulta [Esecuzioni di processi](#) nella Guida per gli sviluppatori di AWS Glue.

- Per API i dettagli, vedere [GetJobRuns](#) in AWS CLI Command Reference.

## get-job

Il seguente esempio di codice mostra come utilizzare `get-job`.

### AWS CLI

Per recuperare le informazioni relative a un processo

L'esempio `get-job` seguente recupera le informazioni relative a un processo.

```

aws glue get-job \
  --job-name my-testing-job

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Processi](#) nella Guida per gli sviluppatori di AWS Glue.

- Per API i dettagli, vedere [GetJob](#) in AWS CLI Command Reference.

## get-plan

Il seguente esempio di codice mostra come utilizzare `get-plan`.

### AWS CLI

Per ottenere il codice generato per la mappatura dei dati dalle tabelle di origine alle tabelle di destinazione

Quanto segue `get-plan` recupera il codice generato per mappare le colonne dall'origine dati alla destinazione dei dati.

```

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"

```

Output:

```

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()
}
}
```

Per ulteriori informazioni, consulta [Modifica degli script in AWS Glue nella Glue Developer Guide](#).AWS

- Per API i dettagli, vedere [GetPlan](#) in AWS CLI Command Reference.

## get-tables

Il seguente esempio di codice mostra come utilizzare `get-tables`.

### AWS CLI

Per elencare le definizioni di alcune o tutte le tabelle del database specificato

L'esempio `get-tables` seguente restituisce le informazioni relative alle tabelle del database specificato.

```
aws glue get-tables --database-name 'tempdb'
```

Output:

```
{
  "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"
}
]
}
```

Per ulteriori informazioni, consulta [Definizione delle tabelle nel AWS Glue Data Catalog](#) nella AWS Glue Developer Guide.

- Per API i dettagli, vedere [GetTables](#) in AWS CLI Command Reference.

## start-crawler

Il seguente esempio di codice mostra come utilizzare `start-crawler`.

### AWS CLI

Per avviare un crawler

L'esempio `start-crawler` seguente avvia un crawler.

```
aws glue start-crawler --name my-crawler
```

Output:

```
None
```

Per ulteriori informazioni, consulta [Definizione di crawler](#) nella Guida per gli sviluppatori di AWS Glue.

- Per API i dettagli, vedere [StartCrawler](#) in AWS CLI Command Reference.

## start-job-run

Il seguente esempio di codice mostra come utilizzare `start-job-run`.

### AWS CLI

Per avviare l'esecuzione di un processo

L'esempio `start-job-run` seguente avvia un processo.

```
aws glue start-job-run \  
  --job-name my-job
```

Output:

```
{  
  "JobRunId":  
  "jr_22208b1f44eb5376a60569d4b21dd20fcb8621e1a366b4e7b2494af764b82ded"  
}
```

Per ulteriori informazioni, consulta [Creazione di processi](#) nella Guida per gli sviluppatori di AWS Glue.

- Per API i dettagli, vedere [StartJobRun](#) in AWS CLI Command Reference.

## GuardDuty esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with GuardDuty.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **accept-invitation**

Il seguente esempio di codice mostra come utilizzare `accept-invitation`.

#### AWS CLI

Accettare un invito a diventare un account GuardDuty membro nella regione corrente

L'esempio seguente mostra come accettare un invito a diventare un account GuardDuty membro nella regione corrente.

```
aws guardduty accept-invitation \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --master-id 123456789111 \  
  --invitation-id d6b94fb03a66ff665f7db8764example
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Gestire GuardDuty gli account tramite invito](#) nella Guida GuardDuty per l'utente.

- Per API i dettagli, vedere [AcceptInvitation](#) in AWS CLI Command Reference.

### **archive-findings**

Il seguente esempio di codice mostra come utilizzare `archive-findings`.

#### AWS CLI

Per archiviare i risultati nella regione corrente

Questo esempio mostra come archiviare i risultati nella regione corrente.

```
aws guardduty archive-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --finding-ids d6b94fb03a66ff665f7db8764example 3eb970e0de00c16ec14e6910fexample
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [la sezione Gestione GuardDuty degli account tramite invito](#) nella Guida GuardDuty per l'utente.

- Per API i dettagli, vedere [ArchiveFindings](#) in AWS CLI Command Reference.

## create-detector

Il seguente esempio di codice mostra come utilizzare `create-detector`.

### AWS CLI

Da abilitare GuardDuty nella regione corrente

Questo esempio mostra come creare un nuovo rilevatore, che abiliti GuardDuty, nella regione corrente. :

```
aws guardduty create-detector \  
  --enable
```

Output:

```
{  
  "DetectorId": "b6b992d6d2f48e64bc59180bfexample"  
}
```

Per ulteriori informazioni, consulta [Enable Amazon GuardDuty](#) nella GuardDuty User Guide.

- Per API i dettagli, consulta [CreateDetector AWS CLI](#) Command Reference.

## create-filter

Il seguente esempio di codice mostra come utilizzare `create-filter`.

### AWS CLI

Per creare un nuovo filtro per la regione corrente

Questo esempio crea un filtro che corrisponde a tutti i risultati di portscan, ad esempio creati da un'immagine specifica. :

```
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"]}}}'
```

Output:

```
{  
  "Name": "myFilter"  
}
```

Per ulteriori informazioni, consulta [Filtraggio dei risultati](#) nella Guida per l'GuardDuty utente.

- Per API i dettagli, vedere [CreateFilter](#) in AWS CLI Command Reference.

## create-ip-set

Il seguente esempio di codice mostra come utilizzare `create-ip-set`.

AWS CLI

Per creare un set IP affidabile

L'`create-ip-set` esempio seguente crea e attiva un set IP affidabile nella regione corrente.

```
aws guardduty create-ip-set \  
--detector-id 12abc34d567e8fa901bc2d34eexample \  
--name new-ip-set \  
--format TXT \  
--location s3://AWSDOC-EXAMPLE-BUCKET/customtrustlist.csv \  
--activate
```

Output:

```
{  
  "IpSetId": "d4b94fc952d6912b8f3060768example"  
}
```

Per ulteriori informazioni, vedere [Utilizzo degli elenchi di IP affidabili e degli elenchi di minacce](#) nella Guida per l' GuardDuty utente.



- Per API i dettagli, vedere [CreatepSetin AWS CLI Command Reference](#).

## create-members

Il seguente esempio di codice mostra come utilizzare `create-members`.

### AWS CLI

Per associare un nuovo membro al tuo account GuardDuty principale nella regione corrente.

Questo esempio mostra come associare gli account dei membri che devono essere gestiti dall'account corrente come GuardDuty master.

```
aws guardduty create-members
  --detector-id b6b992d6d2f48e64bc59180bfexample \
  --account-details AccountId=111122223333,Email=first
+member@example.com AccountId=111111111111 ,Email=another+member@example.com
```

Output:

```
{
  "UnprocessedAccounts": []
}
```

Per ulteriori informazioni, consulta [Gestione di più account](#) nella Guida GuardDuty per l'utente.

- Per API i dettagli, vedere [CreateMembersin AWS CLI Command Reference](#).

## create-publishing-destination

Il seguente esempio di codice mostra come utilizzare `create-publishing-destination`.

### AWS CLI

Per creare una destinazione di pubblicazione in cui esportare i GuardDuty risultati nella regione corrente.

Questo esempio mostra come creare una destinazione di pubblicazione per GuardDuty i risultati.

```
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
```

Output:

```
{  
  "DestinationId": "46b99823849e1bbc242dfbe3cexample"  
}
```

Per ulteriori informazioni, consulta [Esportazione dei risultati](#) nella Guida per l'GuardDuty utente.

- Per API i dettagli, vedere [CreatePublishingDestination](#) in AWS CLI Command Reference.

## create-sample-findings

Il seguente esempio di codice mostra come utilizzare `create-sample-findings`.

AWS CLI

Per creare GuardDuty risultati di esempio nella regione corrente.

Questo esempio mostra come creare un esempio di ricerca dei tipi forniti.

```
aws guardduty create-sample-findings \  
--detector-id b6b992d6d2f48e64bc59180bfexample \  
--finding-types UnauthorizedAccess:EC2/TorClient UnauthorizedAccess:EC2/TorRelay
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Esempi di risultati](#) nella Guida GuardDuty per l'utente.

- Per API i dettagli, vedere [CreateSampleFindings](#) in AWS CLI Command Reference.

## create-threat-intel-set

Il seguente esempio di codice mostra come utilizzare `create-threat-intel-set`.

AWS CLI

Per creare un nuovo set di informazioni sulle minacce nella regione corrente.

Questo esempio mostra come caricare un set di informazioni sulle minacce GuardDuty e attivarlo immediatamente.

```
aws guardduty create-threat-intel-set \  
  --detector-id b6b992d6d2f48e64bc59180bfexample \  
  --name myThreatSet \  
  --format TXT \  
  --location s3://EXAMPLEBUCKET/threatlist.csv \  
  --activate
```

Output:

```
{  
  "ThreatIntelSetId": "20b9a4691aeb33506b808878cexample"  
}
```

Per ulteriori informazioni, consulta [IP attendibili ed elenchi di minacce](#) nella Guida GuardDuty per l'utente.

- Per API i dettagli, vedere [CreateThreatIntelSet](#) in AWS CLI Command Reference.

## decline-invitations

Il seguente esempio di codice mostra come utilizzare `decline-invitations`.

AWS CLI

Per rifiutare un invito a far gestire Guardduty da un altro account nella regione corrente.

Questo esempio mostra come rifiutare un invito all'iscrizione.

```
aws guardduty decline-invitations \  
  --account-ids 111122223333
```

Output:

```
{  
  "UnprocessedAccounts": []  
}
```

Per ulteriori informazioni, consulta [Gestione GuardDuty degli account su invito](#) nella Guida GuardDuty per l'utente.

- Per API i dettagli, vedere [DeclineInvitations](#) in AWS CLI Command Reference.

## delete-detector

Il seguente esempio di codice mostra come utilizzare `delete-detector`.

### AWS CLI

Per eliminare e disattivare GuardDuty un rilevatore nella regione corrente.

Questo esempio mostra come eliminare un rilevatore. In caso di successo, questa operazione verrà disattivata GuardDuty nella regione associata a quel rilevatore.

```
aws guardduty delete-detector \  
  --detector-id b6b992d6d2f48e64bc59180bfexample
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consultate [Sospensione o disabilitazione GuardDuty](#) nella Guida per l'utente. GuardDuty

- Per API i dettagli, vedere [DeleteDetector](#) in AWS CLI Command Reference.

## delete-filter

Il seguente esempio di codice mostra come utilizzare `delete-filter`.

### AWS CLI

Per eliminare un filtro esistente nella regione corrente

Questo esempio mostra come creare ed eliminare un filtro.

```
aws guardduty delete-filter \  
  --detector-id b6b992d6d2f48e64bc59180bfexample \  
  --filter-name byebyeFilter
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Filtraggio dei risultati](#) nella Guida per l' GuardDuty utente.

- Per API i dettagli, vedere [DeleteFilter](#) in AWS CLI Command Reference.

## **disable-organization-admin-account**

Il seguente esempio di codice mostra come utilizzare `disable-organization-admin-account`.

### AWS CLI

Per rimuovere un account come amministratore delegato all' GuardDuty interno dell'organizzazione

Questo esempio mostra come rimuovere un account come amministratore delegato di GuardDuty

```
aws guardduty disable-organization-admin-account \  
--admin-account-id 111122223333
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Gestire gli account con AWS le organizzazioni](#) nella Guida per l'GuardDuty utente.

- Per API i dettagli, vedere [DisableOrganizationAdminAccount](#) in AWS CLI Command Reference.

## **disassociate-from-master-account**

Il seguente esempio di codice mostra come utilizzare `disassociate-from-master-account`.

### AWS CLI

Per dissociarsi dal proprio account principale corrente nella regione corrente

L'`disassociate-from-master-account` esempio seguente dissocia il tuo account dall'account GuardDuty principale corrente nella regione corrente. AWS

```
aws guardduty disassociate-from-master-account \  
--detector-id d4b040365221be2b54a6264dcexample
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Comprensione della relazione tra account GuardDuty principale e account membro](#) nella Guida per l' GuardDuty utente.

- Per API i dettagli, vedere [DisassociateFromMasterAccount](#) in AWS CLI Command Reference.

## get-detector

Il seguente esempio di codice mostra come utilizzare `get-detector`.

### AWS CLI

Per recuperare i dettagli di un rilevatore specifico

L'`get-detector` esempio seguente visualizza i dettagli delle configurazioni del rilevatore specificato.

```
aws guardduty get-detector \  
  --detector-id 12abc34d567e8fa901bc2d34eexample
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, vedere [Concetti e terminologia](#) nella Guida per l'utente GuardDuty.

- Per API i dettagli, vedere [GetDetector](#) in AWS CLI Command Reference.

## get-findings

Il seguente esempio di codice mostra come utilizzare `get-findings`.

### AWS CLI

Esempio 1: Per recuperare i dettagli di un risultato specifico

L'`get-findings` esempio seguente recupera i dettagli completi del JSON risultato specificato.

```
aws guardduty get-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --finding-id 1ab92989eaf0e742df4a014d5example
```

## Output:

```
{
  "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"
}
]
}

```

Per ulteriori informazioni, vedere [Findings](#) nella Guida per l' GuardDuty utente.

- Per API i dettagli, vedere [GetFindings](#) in AWS CLI Command Reference.

## get-ip-set

Il seguente esempio di codice mostra come utilizzare `get-ip-set`.

### AWS CLI

Per elencare, ottieni i dettagli su un set IP affidabile specificato

L'`get-ip-set` esempio seguente mostra lo stato e i dettagli del set IP affidabile specificato.

```

aws guardduty get-ip-set \
  --detector-id 12abc34d567e8fa901bc2d34eexample \
  --ip-set-id d4b94fc952d6912b8f3060768example

```



**Output:**

```
{
  "Status": "ACTIVE",
  "Location": "s3://AWSDOC-EXAMPLE-BUCKET.s3-us-west-2.amazonaws.com/
customlist.csv",
  "Tags": {},
  "Format": "TXT",
  "Name": "test-ip-set"
}
```

Per ulteriori informazioni, vedere [Utilizzo degli elenchi di IP affidabili e degli elenchi di minacce nella Guida per l' GuardDuty utente](#).

- Per API i dettagli, vedere [GetIpSetin AWS CLI Command Reference](#).

**get-master-account**

Il seguente esempio di codice mostra come utilizzare `get-master-account`.

**AWS CLI**

Per recuperare i dettagli sul tuo account principale nella regione corrente

L'`get-master-account`esempio seguente mostra lo stato e i dettagli dell'account principale associato al rilevatore nella regione corrente.

```
aws guardduty get-master-account \
  --detector-id 12abc34d567e8fa901bc2d34eexample
```

**Output:**

```
{
  "Master": {
    "InvitationId": "04b94d9704854a73f94e061e8example",
    "InvitedAt": "2020-06-09T22:23:04.970Z",
    "RelationshipStatus": "Enabled",
    "AccountId": "123456789111"
  }
}
```

Per ulteriori informazioni, consulta [Comprensione della relazione tra account GuardDuty principale e account membro](#) nella Guida per l' GuardDuty utente.

- Per API i dettagli, vedere [GetMasterAccount](#) in AWS CLI Command Reference.

## list-detectors

Il seguente esempio di codice mostra come utilizzare `list-detectors`.

### AWS CLI

Per elencare i rilevatori disponibili nella regione corrente

L'`list-detectors` esempio seguente elenca i rilevatori disponibili nella regione corrente AWS .

```
aws guardduty list-detectors
```

Output:

```
{
  "DetectorIds": [
    "12abc34d567e8fa901bc2d34eexample"
  ]
}
```

Per ulteriori informazioni, vedere [Concetti e terminologia nella Guida](#) per l' GuardDuty utente.

- Per API i dettagli, vedere [ListDetectors](#) in AWS CLI Command Reference.

## list-findings

Il seguente esempio di codice mostra come utilizzare `list-findings`.

### AWS CLI

Esempio 1: Per elencare tutti i risultati per la regione corrente

L'`list-findings` esempio seguente visualizza un elenco di tutti i dati `findingIds` per la regione corrente, ordinati per gravità, dal più alto al più basso.

```
aws guardduty list-findings \
```

```
--detector-id 12abc34d567e8fa901bc2d34eexample \  
--sort-criteria '{"AttributeName": "severity", "OrderBy": "DESC"}'
```

Output:

```
{  
  "FindingIds": [  
    "04b8ab50fd29c64fc771b232dexample",  
    "5ab8ab50fd21373735c826d3aexample",  
    "90b93de7aba69107f05bbe60bexample",  
    ...  
  ]  
}
```

Per ulteriori informazioni, vedere [Findings](#) in the GuardDuty User Guide.

Esempio 2: per elencare i risultati per la regione corrente che corrispondono a criteri di ricerca specifici

L'`list-findings` seguente visualizza un elenco di tutti i risultati `findingIds` che corrispondono a un tipo di risultato specificato.

```
aws guardduty list-findings \  
  --detector-id 12abc34d567e8fa901bc2d34eexample \  
  --finding-criteria '{"Criterion":{"type": {"Eq":["UnauthorizedAccess:EC2/  
SSHBruteForce"]}}}'
```

Output:

```
{  
  "FindingIds": [  
    "90b93de7aba69107f05bbe60bexample",  
    "6eb9430d7023d30774d6f05e3example",  
    "2eb91a2d060ac9a21963a5848example",  
    "44b8ab50fd2b0039a9e48f570example",  
    "9eb8ab4cd2b7e5b66ba4f5e96example",  
    "e0b8ab3a38e9b0312cc390ceeexample"  
  ]  
}
```

Per ulteriori informazioni, vedere [Findings](#) nella Guida GuardDuty per l'utente.

Esempio 3: per elencare i risultati per la regione corrente che corrispondono a una serie specifica di criteri di ricerca definiti all'interno di un JSON file

L'`list-findings` seguente visualizza un elenco di tutto ciò `findingIds` che non è archiviato e coinvolge l'IAMutente denominato «testuser», come specificato in un JSON file.

```
aws guardduty list-findings \
  --detector-id 12abc34d567e8fa901bc2d34eexample \
  --finding-criteria file://myfile.json
```

Contenuto di `myfile.json`.

```
{
  "Criterion": {
    "resource.accessKeyDetails.userName": {
      "Eq": [
        "testuser"
      ]
    },
    "service.archived": {
      "Eq": [
        "false"
      ]
    }
  }
}
```

Output:

```
{
  "FindingIds": [
    "1ab92989eaf0e742df4a014d5example"
  ]
}
```

Per ulteriori informazioni, vedere [Findings](#) in the GuardDuty User Guide.

- Per API i dettagli, vedere [ListFindings](#) in AWS CLI Command Reference.

## list-invitations

Il seguente esempio di codice mostra come utilizzare `list-invitations`.

## AWS CLI

Per elencare i dettagli dei tuoi inviti a diventare un account membro nella regione corrente

L'`list-invitations` seguente elenca i dettagli e lo stato degli inviti a diventare un account GuardDuty membro nella regione corrente.

```
aws guardduty list-invitations
```

Output:

```
{
  "Invitations": [
    {
      "InvitationId": "d6b94fb03a66ff665f7db8764example",
      "InvitedAt": "2020-06-10T17:56:38.221Z",
      "RelationshipStatus": "Invited",
      "AccountId": "123456789111"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Gestire GuardDuty gli account tramite invito nella Guida per l'utente](#).

- Per API i dettagli, vedere [ListInvitations](#) in AWS CLI Command Reference.

## list-ip-sets

Il seguente esempio di codice mostra come utilizzare `list-ip-sets`.

### AWS CLI

Per elencare i set IP affidabili nella regione corrente

L'`list-ip-sets` seguente elenca i set IP affidabili nella AWS regione corrente.

```
aws guardduty list-ip-sets \
  --detector-id 12abc34d567e8fa901bc2d34eexample
```

Output:

```
{
  "IpSetIds": [
    "d4b94fc952d6912b8f3060768example"
  ]
}
```

Per ulteriori informazioni, vedere [Utilizzo degli elenchi di IP affidabili e degli elenchi di minacce](#) nella Guida GuardDuty per l'utente.

- Per API i dettagli, vedere [ListIpSets](#) in AWS CLI Command Reference.

## list-members

Il seguente esempio di codice mostra come utilizzare `list-members`.

### AWS CLI

Per elencare tutti i membri della regione corrente

L'`list-members` esempio seguente elenca tutti gli account membro e i relativi dettagli per l'area corrente.

```
aws guardduty list-members \
  --detector-id 12abc34d567e8fa901bc2d34eexample
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Comprensione della relazione tra account GuardDuty principale e account membro](#) nella Guida per l'utente GuardDuty.

- Per API i dettagli, vedere [ListMembers](#) in AWS CLI Command Reference.

## update-ip-set

Il seguente esempio di codice mostra come utilizzare `update-ip-set`.

### AWS CLI

Per aggiornare un set IP affidabile

L'esempio seguente mostra come aggiornare i dettagli di un set IP affidabile.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Utilizzo degli elenchi di IP affidabili e degli elenchi di minacce](#) nella Guida GuardDuty per l'utente.

- Per API i dettagli, vedere [UpdateIpSet](#) in AWS CLI Command Reference.

## AWS Health esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Health.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **describe-affected-entities**

Il seguente esempio di codice mostra come utilizzare `describe-affected-entities`.

#### AWS CLI

Per elencare le entità interessate da uno specifico evento AWS Health

L'esempio seguente elenca le entità interessate dall'evento AWS Health specificato. Questo evento è una notifica di fatturazione per l'AWS account.

```
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
```

#### Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta [Event log](#) nella AWS Health User Guide.

- Per API i dettagli, vedere [DescribeAffectedEntities](#) in AWS CLI Command Reference.

### **describe-event-details**

Il seguente esempio di codice mostra come utilizzare `describe-event-details`.



## AWS CLI

Per elencare informazioni su un evento AWS Health

L'`describe-event-details`sempio seguente elenca le informazioni sull'evento AWS Health specificato.

```
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
```

Output:

```
{  
  "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": []  
}
```

Per ulteriori informazioni, consulta il [riquadro dei dettagli dell'evento](#) nella AWS Health User Guide.

- Per API i dettagli, vedere [DescribeEventDetails](#) in AWS CLI Command Reference.

## describe-events

Il seguente esempio di codice mostra come utilizzare `describe-events`.

### AWS CLI

Esempio 1: elencare gli eventi AWS Health

L'`describe-events` esempio seguente elenca gli eventi AWS Health recenti.

```
aws health describe-events \  
  --region us-east-1
```

Output:

```
{  
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta la [Guida introduttiva alla AWS Personal Health Dashboard](#) nella AWS Health User Guide.

Esempio 2: elencare gli eventi AWS Health per servizio e codice di stato dell'evento

L'`describe-events`esempio seguente elenca gli eventi AWS Health per Amazon Elastic Compute Cloud (AmazonEC2) in cui lo stato dell'evento è chiuso.

```

aws health describe-events \
  --filter "services=EC2,eventStatusCodes=closed"

```

Output:

```

{
  "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_COBJ_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"
    }
]
}

```

Per ulteriori informazioni, consulta la [Guida introduttiva alla AWS Personal Health Dashboard](#) nella AWS Health User Guide.

- Per API i dettagli, vedere [DescribeEvents](#) in AWS CLI Command Reference.

## HealthImaging esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with HealthImaging.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **copy-image-set**

Il seguente esempio di codice mostra come utilizzare `copy-image-set`.

AWS CLI

Esempio 1: copiare un set di immagini senza una destinazione.

L'`copy-image-set` esempio seguente crea una copia duplicata di un set di immagini senza una destinazione.

```
aws medical-imaging copy-image-set \  
  --datastore-id 12345678901234567890123456789012 \  
  --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
  --copy-image-set-information '{"sourceImageSet": {"latestVersionId": "1" } }'
```

Output:

```
{  
  "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"
}

```

Esempio 2: copiare un set di immagini con una destinazione.

L'copy-image-setesempio seguente crea una copia duplicata di un set di immagini con una destinazione.

```

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"} }'

```

Output:

```

{
  "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"
}

```



}

Esempio 3: copiare un sottoinsieme di istanze da un set di immagini di origine a un set di immagini di destinazione.

L'copy-image-setesempio seguente copia un'DICOMistanza dal set di immagini di origine al set di immagini di destinazione. Il parametro force viene fornito per ignorare le incongruenze negli attributi dei livelli Patient, Study e Series.

```
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
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Copiare un set di immagini](#) nella Guida per gli sviluppatori AWS HealthImaging

- Per API i dettagli, vedere [CopyImageSetin AWS CLI Command Reference](#).

## create-datastore

Il seguente esempio di codice mostra come utilizzare `create-datastore`.

### AWS CLI

Per creare un archivio dati

Il seguente esempio di `create-datastore` codice crea un archivio dati con il nome `my-datastore`.

```
aws medical-imaging create-datastore \  
  --datastore-name "my-datastore"
```

Output:

```
{  
  "datastoreId": "12345678901234567890123456789012",  
  "datastoreStatus": "CREATING"  
}
```

Per ulteriori informazioni, consulta [Creazione di un data store](#) nella Guida per gli AWS HealthImaging sviluppatori.

- Per API i dettagli, consulta [CreateDatastore AWS CLI Command Reference](#).

## delete-datastore

Il seguente esempio di codice mostra come utilizzare `delete-datastore`.

### AWS CLI

Per eliminare un archivio dati

Il seguente esempio di `delete-datastore` codice elimina un data store.

```
aws medical-imaging delete-datastore \  
--datastore-id "12345678901234567890123456789012"
```

Output:

```
{  
  "datastoreId": "12345678901234567890123456789012",  
  "datastoreStatus": "DELETING"  
}
```

Per ulteriori informazioni, consulta [Eliminazione di un data store nella Guida](#) per gli AWS HealthImaging sviluppatori.

- Per API i dettagli, consulta [DeleteDatastore AWS CLI Command Reference](#).

## delete-image-set

Il seguente esempio di codice mostra come utilizzare `delete-image-set`.

AWS CLI

Per eliminare un set di immagini

Il seguente esempio di `delete-image-set` codice elimina un set di immagini.

```
aws medical-imaging delete-image-set \  
--datastore-id 12345678901234567890123456789012 \  
--image-set-id ea92b0d8838c72a3f25d00d13616f87e
```

Output:

```
{  
  "imageSetWorkflowStatus": "DELETING",  
  "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",  
  "imageSetState": "LOCKED",  
  "datastoreId": "12345678901234567890123456789012"  
}
```

Per ulteriori informazioni, consultate [Eliminazione di un set di immagini nella Guida](#) per gli AWS HealthImaging sviluppatori.

- Per API i dettagli, vedere [DeleteImageSet](#) in AWS CLI Command Reference.

## get-datastore

Il seguente esempio di codice mostra come utilizzare `get-datastore`.

### AWS CLI

Per ottenere le proprietà di un archivio dati

Il seguente esempio di `get-datastore` codice ottiene le proprietà di un data store.

```
aws medical-imaging get-datastore \
  --datastore-id 12345678901234567890123456789012
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Ottenere le proprietà del data store](#) nella Guida per gli AWS HealthImaging sviluppatori.

- Per API i dettagli, vedere [GetDatastore](#) in AWS CLI Command Reference.

## get-dicom-import-job

Il seguente esempio di codice mostra come utilizzare `get-dicom-import-job`.

### AWS CLI

Per ottenere le proprietà di un processo di importazione dicom

Il seguente esempio di `get-dicom-import-job` codice ottiene le proprietà di un processo di importazione dicom.

```
aws medical-imaging get-dicom-import-job \  
  --datastore-id "12345678901234567890123456789012" \  
  --job-id "09876543210987654321098765432109"
```

Output:

```
{  
  "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/"  
  }  
}
```

Per ulteriori informazioni, consulta [Ottenere le proprietà del processo di importazione](#) nella Guida per gli AWS HealthImaging sviluppatori.

- Per API i dettagli, vedere [GetDICOMImport Job](#) in AWS CLI Command Reference.

## get-image-frame

Il seguente esempio di codice mostra come utilizzare `get-image-frame`.

### AWS CLI

Per ottenere i dati dei pixel impostati dall'immagine

Il seguente esempio di `get-image-frame` codice ottiene una cornice di immagine.

```
aws medical-imaging get-image-frame \  

```

```
--datastore-id "12345678901234567890123456789012" \  
--image-set-id "98765412345612345678907890789012" \  
--image-frame-information imageFrameId=3abf5d5d7ae72f80a0ec81b2c0de3ef4 \  
imageframe.jpg
```

Nota: questo esempio di codice non include l'output perché l' `GetImageFrame` azione restituisce un flusso di dati di pixel al file `imageframe.jpg`. Per informazioni sulla decodifica e la visualizzazione dei frame di immagini, consultate le librerie di decodifica. HTJ2K

Per ulteriori informazioni, consultate [Ottenere i dati dei pixel del set di immagini](#) nella Guida per gli AWS HealthImaging sviluppatori.

- Per API i dettagli, consultate [GetImageFrame AWS CLI Command Reference](#).

## get-image-set-metadata

Il seguente esempio di codice mostra come utilizzare `get-image-set-metadata`.

### AWS CLI

Esempio 1: Per ottenere i metadati del set di immagini senza versione

Il seguente esempio di `get-image-set-metadata` codice ottiene i metadati per un set di immagini senza specificare una versione.

Nota: `outfile` è un parametro obbligatorio

```
aws medical-imaging get-image-set-metadata \  
--datastore-id 12345678901234567890123456789012 \  
--image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
studymetadata.json.gz
```

I metadati restituiti vengono compressi con `gzip` e archiviati nel file `studymetadata.json.gz`. Per visualizzare il contenuto dell'oggetto restituito, è necessario prima decomprimerlo. JSON

Output:

```
{  
  "contentType": "application/json",  
  "contentEncoding": "gzip"  
}
```

## Esempio 2: per ottenere i metadati del set di immagini con la versione

Il seguente esempio di `get-image-set-metadata` codice ottiene i metadati per un set di immagini con una versione specificata.

Nota: `outfile` è un parametro obbligatorio

```
aws medical-imaging get-image-set-metadata \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e \  
  --version-id 1 \  
  studymetadata.json.gz
```

I metadati restituiti vengono compressi con `gzip` e archiviati nel file `studymetadata.json.gz`. Per visualizzare il contenuto dell'oggetto restituito, è necessario prima decomprimerlo. JSON

Output:

```
{  
  "contentType": "application/json",  
  "contentEncoding": "gzip"  
}
```

Per ulteriori informazioni, consulta [Ottenere i metadati del set di immagini](#) nella Guida per gli AWS HealthImaging sviluppatori.

- Per API i dettagli, consulta [GetImageSetMetadata AWS CLI Command Reference](#).

## get-image-set

Il seguente esempio di codice mostra come utilizzare `get-image-set`.

AWS CLI

Per ottenere le proprietà del set di immagini

Il seguente esempio di `get-image-set` codice ottiene le proprietà di un set di immagini.

```
aws medical-imaging get-image-set \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id 18f88ac7870584f58d56256646b4d92b \  
  outfile
```

```
--version-id 1
```

Output:

```
{
  "versionId": "1",
  "imageSetWorkflowStatus": "COPIED",
  "updatedAt": 1680027253.471,
  "imageSetId": "18f88ac7870584f58d56256646b4d92b",
  "imageSetState": "ACTIVE",
  "createdAt": 1679592510.753,
  "datastoreId": "12345678901234567890123456789012"
}
```

Per ulteriori informazioni, consulta [Ottenere le proprietà del set di immagini](#) nella Guida per gli AWS HealthImaging sviluppatori.

- Per API i dettagli, vedere [GetImageSet](#) in AWS CLI Command Reference.

## list-datastores

Il seguente esempio di codice mostra come utilizzare `list-datastores`.

AWS CLI

Per elencare gli archivi dati

Il seguente esempio di `list-datastores` codice elenca gli archivi dati disponibili.

```
aws medical-imaging list-datastores
```

Output:

```
{
  "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"
      }
    ]
  }
```

Per ulteriori informazioni, consulta [Elencare gli archivi di dati](#) nella Guida per gli AWS HealthImaging sviluppatori.

- Per API i dettagli, consulta [ListDatastores AWS CLI Command Reference](#).

## list-dicom-import-jobs

Il seguente esempio di codice mostra come utilizzare `list-dicom-import-jobs`.

### AWS CLI

Per elencare i lavori di importazione dicom

Il seguente esempio di `list-dicom-import-jobs` codice elenca i processi di importazione dicom.

```
aws medical-imaging list-dicom-import-jobs \
  --datastore-id "12345678901234567890123456789012"
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Elencare i lavori di importazione](#) nella AWS HealthImaging Developer Guide.

- Per API i dettagli, consulta [ListDICOMImport Jobs](#) in AWS CLI Command Reference.

## list-image-set-versions

Il seguente esempio di codice mostra come utilizzare `list-image-set-versions`.

### AWS CLI

Per elencare le versioni dei set di immagini

Il seguente esempio di `list-image-set-versions` codice elenca la cronologia delle versioni di un set di immagini.

```
aws medical-imaging list-image-set-versions \  
  --datastore-id 12345678901234567890123456789012 \  
  --image-set-id ea92b0d8838c72a3f25d00d13616f87e
```

Output:

```
{  
  "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
    }
]
}

```

Per ulteriori informazioni, consultate [Elenco delle versioni dei set di immagini](#) nella Guida per gli AWS HealthImaging sviluppatori.

- Per API i dettagli, consulta [ListImageSetVersions AWS CLI Command Reference](#).

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Esempio 1: per elencare i tag delle risorse per un archivio dati

Il seguente esempio di `list-tags-for-resource` codice elenca i tag per un data store.

```

aws medical-imaging list-tags-for-resource \
  --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012"

```

Output:

```

{
  "tags":{
    "Deployment":"Development"
  }
}

```

```
}  
}
```

Esempio 2: Elencare i tag delle risorse per un set di immagini

Il seguente esempio di `list-tags-for-resource` codice elenca i tag per un set di immagini.

```
aws medical-imaging list-tags-for-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/1234567890123456789012/3456789012/  
imageset/18f88ac7870584f58d56256646b4d92b"
```

Output:

```
{  
  "tags":{  
    "Deployment":"Development"  
  }  
}
```

Per ulteriori informazioni, consulta [Tagging resources with AWS HealthImaging](#) nella AWS HealthImaging Developer Guide.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## search-image-sets

Il seguente esempio di codice mostra come utilizzare `search-image-sets`.

AWS CLI

Esempio 1: Per cercare set di immagini con un EQUAL operatore

Il seguente esempio di `search-image-sets` codice utilizza l'EQUAL operatore per cercare set di immagini in base a un valore specifico.

```
aws medical-imaging search-image-sets \  
  --datastore-id 12345678901234567890123456789012 \  
  --search-criteria file://search-criteria.json
```

Contenuto di `search-criteria.json`

```
{
  "filters": [{
    "values": [{"DICOMPatientId" : "SUBJECT08701"}],
    "operator": "EQUAL"
  }]
}
```

Output:

```
{
  "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"
  }]
}
```

Esempio 2: Per cercare set di immagini con un BETWEEN operatore utilizzando DICOMStudyDate e DICOMStudyTime

Il seguente esempio di `search-image-sets` codice cerca set di immagini con DICOM studi generati tra il 1° gennaio 1990 (00:00) e il 1° gennaio 2023 (00:00).

Nota: DICOMStudyTime è facoltativo. Se non è presente, 12:00 AM (inizio giornata) è il valore temporale per le date fornite per il filtraggio.

```
aws medical-imaging search-image-sets \
  --datastore-id 12345678901234567890123456789012 \
```

```
--search-criteria file://search-criteria.json
```

## Contenuto di search-criteria.json

```
{
  "filters": [{
    "values": [{
      "DICOMStudyDateAndTime": {
        "DICOMStudyDate": "19900101",
        "DICOMStudyTime": "000000"
      }
    },
    {
      "DICOMStudyDateAndTime": {
        "DICOMStudyDate": "20230101",
        "DICOMStudyTime": "000000"
      }
    }
  ]},
  "operator": "BETWEEN"
}]
}
```

## Output:

```
{
  "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"
  ]
}
```

```
    ]]
  }
```

Esempio 3: per cercare set di immagini con un BETWEEN operatore utilizzando createdAt (gli studi temporali erano precedentemente persistenti)

Il seguente esempio di `search-image-sets` codice cerca set di immagini con DICOM gli studi persistenti HealthImaging tra gli intervalli di tempo del fuso orario. UTC

Nota: fornite un createdAt formato di esempio («1985-04-12T 23:20:50.52 Z»).

```
aws medical-imaging search-image-sets \
  --datastore-id 12345678901234567890123456789012 \
  --search-criteria file://search-criteria.json
```

Contenuto di `search-criteria.json`

```
{
  "filters": [{
    "values": [{
      "createdAt": "1985-04-12T23:20:50.52Z"
    },
    {
      "createdAt": "2022-04-12T23:20:50.52Z"
    }
  ]],
  "operator": "BETWEEN"
}]
}
```

Output:

```
{
  "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"
}]
}

```

Esempio 4: per cercare set di immagini con un EQUAL operatore attivo e attivo DICOMSeriesInstanceUID e ordinare le risposte in ordine sul campo BETWEEN updatedAt ASC updatedAt

Il seguente esempio di search-image-sets codice cerca i set di immagini con un EQUAL operatore on DICOMSeriesInstanceUID e BETWEEN on updatedAt e ordina la risposta in ASC ordine sul updatedAt campo.

Nota: fornite un updatedAt formato di esempio («1985-04-12T 23:20:50.52 Z»).

```

aws medical-imaging search-image-sets \
  --datastore-id 12345678901234567890123456789012 \
  --search-criteria file://search-criteria.json

```

Contenuto di 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"
}
}
```

Output:

```
{
  "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"
  ]
}
```

[Per ulteriori informazioni, consulta \*Searching image sets nella Developer Guide.AWS HealthImaging\*](#)

- Per API i dettagli, vedere [SearchImageSets](#) in AWS CLI Command Reference.

## **start-dicom-import-job**

Il seguente esempio di codice mostra come utilizzare `start-dicom-import-job`.

AWS CLI

Per avviare un processo di importazione dicom

Il seguente esempio di `start-dicom-import-job` codice avvia un processo di importazione dicom.

```
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"
```

Output:

```
{  
  "datastoreId": "12345678901234567890123456789012",  
  "jobId": "09876543210987654321098765432109",  
  "jobStatus": "SUBMITTED",  
  "submittedAt": "2022-08-12T11:28:11.152000+00:00"  
}
```

Per ulteriori informazioni, consulta [Avvio di un processo di importazione](#) nella Guida per gli AWS HealthImaging sviluppatori.

- Per API i dettagli, vedere [StartDICOMImport Job](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

AWS CLI

Esempio 1: etichettare un archivio dati

I seguenti esempi di `tag-resource` codice contrassegnano un data store.

```
aws medical-imaging tag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012" \  
  --tags '{"Deployment":"Development"}'
```

Questo comando non produce alcun output.

## Esempio 2: etichettare un set di immagini

I seguenti esempi di `tag-resource` codice contrassegnano un set di immagini.

```
aws medical-imaging tag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012/  
imageset/18f88ac7870584f58d56256646b4d92b" \  
  --tags '{"Deployment":"Development"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging resources with AWS HealthImaging](#) nella AWS HealthImaging Developer Guide.

- Per API i dettagli, consulta [TagResource AWS CLI](#) Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Esempio 1: rimuovere i tag da un archivio dati

Il seguente esempio di `untag-resource` codice rimuove i tag da un data store.

```
aws medical-imaging untag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012" \  
  --tag-keys ['Deployment']
```

Questo comando non produce alcun output.

Esempio 2: rimuovere i tag da un set di immagini

Il seguente esempio di `untag-resource` codice rimuove i tag da un set di immagini.

```
aws medical-imaging untag-resource \  
  --resource-arn "arn:aws:medical-imaging:us-  
east-1:123456789012:datastore/12345678901234567890123456789012/  
imageset/18f88ac7870584f58d56256646b4d92b" \  
  --tag-keys ['Deployment']
```

```
--tag-keys '["Deployment"]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging resources with AWS HealthImaging](#) nella Developer Guide.AWS HealthImaging

- Per API i dettagli, consulta [UntagResource AWS CLI Command Reference](#).

## update-image-set-metadata

Il seguente esempio di codice mostra come utilizzare `update-image-set-metadata`.

### AWS CLI

Esempio 1: Inserire o aggiornare un attributo nei metadati del set di immagini

L'`update-image-set-metadata`esempio seguente inserisce o aggiorna un attributo nei metadati del set di immagini.

```
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
```

### Contenuto di metadata-updates.json

```
{  
  "DICOMUpdates": {  
    "updatableAttributes": "{\"SchemaVersion\":1.1,\"Patient\":{\"DICOM\":  
    {\"PatientName\": \"MX^MX\"}}}"  
  }  
}
```

### Output:

```
{  
  "latestVersionId": "2",  
  "imageSetWorkflowStatus": "UPDATING",
```

```

    "updatedAt": 1680042257.908,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
  }

```

## Esempio 2: rimuovere un attributo dai metadati del set di immagini

L'update-image-set-metadata esempio seguente rimuove un attributo dai metadati del set di immagini.

```

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

```

## Contenuto di metadata-updates.json

```

{
  "DICOMUpdates": {
    "removableAttributes": "{\"SchemaVersion\":1.1,\"Study\":{\"DICOM\":{\"StudyDescription\":\"CHEST\"}}}"
  }
}

```

## Output:

```

{
  "latestVersionId": "2",
  "imageSetWorkflowStatus": "UPDATING",
  "updatedAt": 1680042257.908,
  "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
  "imageSetState": "LOCKED",
  "createdAt": 1680027126.436,
  "datastoreId": "12345678901234567890123456789012"
}

```

## Esempio 3: rimuovere un'istanza dai metadati del set di immagini

L'update-image-set-metadata esempio seguente rimuove un'istanza dai metadati del set di immagini.

```
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
```

Contenuto di 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\": {}}}}}}\"  
  }  
}
```

Output:

```
{  
  "latestVersionId": "2",  
  "imageSetWorkflowStatus": "UPDATING",  
  "updatedAt": 1680042257.908,  
  "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",  
  "imageSetState": "LOCKED",  
  "createdAt": 1680027126.436,  
  "datastoreId": "12345678901234567890123456789012"  
}
```

Esempio 4: Per ripristinare una versione precedente di un set di immagini

L'update-image-set-metadata esempio seguente mostra come ripristinare un set di immagini a una versione precedente. CopyImageSet e UpdateImageSetMetadata le azioni creano nuove versioni dei set di immagini.

```
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
```

```
--cli-binary-format raw-in-base64-out \  
--update-image-set-metadata-updates '{"revertToVersionId": "1"}'
```

Output:

```
{  
  "datastoreId": "12345678901234567890123456789012",  
  "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",  
  "latestVersionId": "4",  
  "imageSetState": "LOCKED",  
  "imageSetWorkflowStatus": "UPDATING",  
  "createdAt": 1680027126.436,  
  "updatedAt": 1680042257.908  
}
```

Esempio 5: aggiungere un elemento di DICOM dati privato a un'istanza

L'update-image-set-metadata esempio seguente mostra come aggiungere un elemento privato a un'istanza specificata all'interno di un set di immagini. Lo DICOM standard consente elementi di dati privati per la comunicazione di informazioni che non possono essere contenuti in elementi di dati standard. È possibile creare, aggiornare ed eliminare elementi di dati privati con l'UpdateImageSetMetadata azione.

```
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
```

Contenuto di 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\"}}}}}}}"  
  }  
}
```

## Output:

```
{
  "latestVersionId": "2",
  "imageSetWorkflowStatus": "UPDATING",
  "updatedAt": 1680042257.908,
  "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
  "imageSetState": "LOCKED",
  "createdAt": 1680027126.436,
  "datastoreId": "12345678901234567890123456789012"
}
```

## Esempio 6: aggiornare un elemento di DICOM dati privato in un'istanza

L'update-image-set-metadata esempio seguente mostra come aggiornare il valore di un elemento di dati privato appartenente a un'istanza all'interno di un set di immagini.

```
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
```

## Contenuto di 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\": {\"00091001\": \"GE_GENESIS_DD\"}}}}}}}"
  }
}
```

## Output:

```
{
  "latestVersionId": "2",
  "imageSetWorkflowStatus": "UPDATING",
```



```

    "updatedAt": 1680042257.908,
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
  }

```

### Esempio 7: Aggiornare a SOPInstanceUID con il parametro force

L'update-image-set-metadata esempio seguente mostra come aggiornare a SOPInstanceUID, utilizzando il parametro force per sovrascrivere i vincoli dei DICOM metadati.

```

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

```

### Contenuto di 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\\\"}}}}}}}"
    }
  }
}

```

### Output:

```

{
  "latestVersionId": "2",
  "imageSetWorkflowStatus": "UPDATING",
  "updatedAt": 1680042257.908,
  "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
  "imageSetState": "LOCKED",
  "createdAt": 1680027126.436,
  "datastoreId": "12345678901234567890123456789012"
}

```

```
}
```

Per ulteriori informazioni, consulta [Aggiornamento dei metadati del set di immagini](#) nella Guida per gli sviluppatori.AWS HealthImaging

- Per API i dettagli, consultate [UpdateImageSetMetadata AWS CLI](#) Command Reference.

## HealthLake esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with HealthLake.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

### Azioni

#### **create-fhir-datastore**

Il seguente esempio di codice mostra come utilizzare `create-fhir-datastore`.

AWS CLI

Per creare un FHIR Data Store.

L'`create-fhir-datastore`esempio seguente mostra come creare un nuovo Data Store in Amazon HealthLake.

```
aws healthlake create-fhir-datastore \  
  --region us-east-1 \  
  --datastore-type-version R4 \  
  --datastore-type-version R4 \  
  --datastore-name "FhirTestDatastore"
```

**Output:**

```
{
  "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)"
}
```

Per ulteriori informazioni, consulta [Creazione e monitoraggio di un FHIR Data Store](#) nell'Amazon HealthLake Developer Guide.

- Per API i dettagli, consulta [CreateFhirDatastore AWS CLI Command Reference](#).

**delete-fhir-datastore**

Il seguente esempio di codice mostra come utilizzare `delete-fhir-datastore`.

**AWS CLI**

Per eliminare un FHIR Data Store

L'`delete-fhir-datastore` esempio seguente mostra come eliminare un Data Store e tutti i suoi contenuti in Amazon HealthLake.

```
aws healthlake delete-fhir-datastore \
  --datastore-id (Data Store ID) \
  --region us-east-1
```

**Output:**

```
{
  "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)"
}
```

Per ulteriori informazioni, consulta [Creazione e monitoraggio di un FHIR Data Store](https://docs.aws.amazon.com/healthlake/latest/devguide/working-with-FHIR-healthlake.html) < <https://docs.aws.amazon.com/healthlake/latest/devguide/working-with-FHIR-healthlake.html>> nella Amazon Developer Guide. HealthLake

- Per API i dettagli, consulta Command Reference. [DeleteFhirDatastore](#) AWS CLI

## describe-fhir-datastore

Il seguente esempio di codice mostra come utilizzare `describe-fhir-datastore`.

### AWS CLI

Per descrivere un FHIR Data Store

L'esempio seguente mostra come trovare le proprietà di un Data Store in Amazon HealthLake.

```
aws healthlake describe-fhir-datastore \
  --datastore-id "1f2f459836ac6c513ce899f9e4f66a59" \
  --region us-east-1
```

Output:

```
{
  "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)"
  }
}
```

Per ulteriori informazioni, consulta [Creazione e monitoraggio di un FHIR Data Store](#) nella Amazon HealthLake Developer Guide.

- Per API i dettagli, consulta [DescribeFhirDatastore AWS CLI Command Reference](#).

## describe-fhir-export-job

Il seguente esempio di codice mostra come utilizzare `describe-fhir-export-job`.

### AWS CLI

Per descrivere un processo di FHIR esportazione

L'esempio seguente mostra come trovare le proprietà di un processo di FHIR esportazione in Amazon HealthLake.

```
aws healthlake describe-fhir-export-job \  
  --datastore-id (Datastore ID) \  
  --job-id 9b9a51943afaedd0a8c0c26c49135a31
```

Output:

```
{  
  "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)"  
  }  
}
```

Per ulteriori informazioni, consulta [Esportazione di file da un FHIR Data Store](#) nell'Amazon HealthLake Developer Guide.

- Per API i dettagli, consulta [DescribeFhirExportJob AWS CLI Command Reference](#).

## describe-fhir-import-job

Il seguente esempio di codice mostra come utilizzare `describe-fhir-import-job`.

## AWS CLI

Per descrivere un processo di FHIR importazione

L'`describe-fhir-import-job` seguente mostra come apprendere le proprietà di un processo di FHIR importazione utilizzando Amazon HealthLake.

```
aws healthlake describe-fhir-import-job \  
  --datastore-id (Datastore ID) \  
  --job-id c145fbb27b192af392f8ce6e7838e34f \  
  --region us-east-1
```

Output:

```
{  
  "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)"  
  }  
}
```

Per ulteriori informazioni, consulta [Importazione di file in un FHIR Data Store](#) nella Amazon HealthLake Developer Guide.

- Per API i dettagli, consulta [DescribeFhirImportJob AWS CLI Command Reference](#).

## **list-fhir-datastores**

Il seguente esempio di codice mostra come utilizzare `list-fhir-datastores`.

## AWS CLI

Per elencare gli archivi FHIR dati

L'`list-fhir-datastore` seguente mostra come utilizzare il comando e come gli utenti possono filtrare i risultati in base allo stato del Data Store in Amazon HealthLake.

```
aws healthlake list-fhir-datastores \  
  --region us-east-1 \  
  --filter DatastoreStatus=ACTIVE
```

Output:

```
{  
  "DatastorePropertiesList": [  
    {  
      "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": "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>"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Creazione e monitoraggio di un FHIR Data Store](#) nell'Amazon HealthLake Developer Guide.

- Per API i dettagli, consulta [ListFhirDatastores AWS CLI Command Reference](#).

## list-fhir-export-jobs

Il seguente esempio di codice mostra come utilizzare `list-fhir-export-jobs`.

### AWS CLI

Per elencare tutti i lavori di FHIR esportazione

L'`list-fhir-export-jobs` esempio seguente mostra come utilizzare il comando per visualizzare un elenco di lavori di esportazione associati a un account.

```
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)
```

Output:

```
{  
  "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
```

Per ulteriori informazioni, consulta [Esportazione di file da un FHIR Data Store](#) nell'Amazon HealthLake Developer Guide.



- Per API i dettagli, consulta [ListFhirExportJobs AWS CLI Command Reference](#).

## list-fhir-import-jobs

Il seguente esempio di codice mostra come utilizzare `list-fhir-import-jobs`.

### AWS CLI

Per elencare tutti i lavori di FHIR importazione

L'`list-fhir-import-jobs` esempio seguente mostra come utilizzare il comando per visualizzare un elenco di tutti i processi di importazione associati a un account.

```
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)
```

Output:

```
{  
  "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
```

Per ulteriori informazioni, consulta [Importazione di file in FHIR Data Store](#) nella Amazon HealthLake Developer Guide.

- Per API i dettagli, consulta [ListFhirImportJobs AWS CLI Command Reference](#).

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag per un Data Store

L'`list-tags-for-resource` esempio seguente elenca i tag associati al Data Store specificato. :

```
aws healthlake list-tags-for-resource \
  --resource-arn "arn:aws:healthlake:us-east-1:674914422125:datastore/
  fhir/0725c83f4307f263e16fd56b6d8ebd8e" \
  --region us-east-1
```

Output:

```
{
  "tags": {
    "key": "value",
    "key1": "value1"
  }
}
```

Per ulteriori informazioni, consulta [Tagging resources in Amazon HealthLake nell'Amazon HealthLake Developer Guide](#).

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## start-fhir-export-job

Il seguente esempio di codice mostra come utilizzare `start-fhir-export-job`.

### AWS CLI

Per avviare un processo di FHIR esportazione

L'`start-fhir-export-job` seguente mostra come avviare un processo di FHIR esportazione utilizzando Amazon HealthLake.

```
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)
```

Output:

```
{  
  "DatastoreId": "(Datastore ID)",  
  "JobStatus": "SUBMITTED",  
  "JobId": "9b9a51943afaedd0a8c0c26c49135a31"  
}
```

Per ulteriori informazioni, consulta [Esportazione di file da un FHIR Data Store](#) nell'Amazon HealthLake Developer Guide.

- Per API i dettagli, consulta [StartFhirExportJob AWS CLI Command Reference](#).

## `start-fhir-import-job`

Il seguente esempio di codice mostra come utilizzare `start-fhir-import-job`.

### AWS CLI

Per avviare un processo di FHIR importazione

L'`start-fhir-import-job` seguente mostra come avviare un processo di FHIR importazione utilizzando Amazon HealthLake.

```
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
```

Output:

```
{
```

```
"DatastoreId": "(Datastore ID)",
"JobStatus": "SUBMITTED",
"JobId": "c145fbb27b192af392f8ce6e7838e34f"
}
```

Per ulteriori informazioni, consulta [Importazione di file in un FHIR Data Store 'https://docs.aws.amazon.com/healthlake/latest/devguide/import-datastore.html'](https://docs.aws.amazon.com/healthlake/latest/devguide/import-datastore.html) nella Amazon HealthLake Developer Guide.

- Per API i dettagli, consulta [StartFhirImportJob AWS CLI Command Reference](#).

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere un tag a Data Store

L'`tag-resource` esempio seguente mostra come aggiungere un tag a un Data Store.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta ['Aggiungere un tag a un Data Store < https://docs.aws.amazon.com/healthlake/latest/devguide/add-a-tag.html>'](https://docs.aws.amazon.com/healthlake/latest/devguide/add-a-tag.html) nella Amazon Developer Guide. HealthLake .

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da un Data Store.

L'untag-resourceesempio seguente mostra come rimuovere i tag da un Data Store.

```
aws healthlake untag-resource \  
  --resource-arn "arn:aws:healthlake:us-east-1:674914422125:datastore/fhir/  
b91723d65c6fdeb1d26543a49d2ed1fa" \  
  --tag-keys '["key1"]' \  
  --region us-east-1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Removing tags from a Data Store](#) nell'Amazon HealthLake Developer Guide.

- Per API i dettagli, consulta [UntagResource AWS CLI Command Reference](#).

## HealthOmics esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with HealthOmics.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **abort-multipart-read-set-upload**

Il seguente esempio di codice mostra come utilizzareabort-multipart-read-set-upload.

AWS CLI

Per interrompere il caricamento di un set di lettura in più parti

L'`abort-multipart-read-set-upload` seguente interrompe il caricamento di un set di lettura in più parti nel vostro archivio di HealthOmics sequenze.

```
aws omics abort-multipart-read-set-upload \  
  --sequence-store-id 0123456789 \  
  --upload-id 1122334455
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consultate [Caricamento diretto su un archivio di sequenze nella Guida per l'AWS HealthOmics utente](#).

- Per API i dettagli, consultate [AbortMultipartReadSetUpload AWS CLI Command Reference](#).

## accept-share

Il seguente esempio di codice mostra come utilizzare `accept-share`.

### AWS CLI

Per accettare una condivisione dei dati di analisi, archivia i dati

L'`accept-share` seguente accetta una condivisione di dati dell'archivio di HealthOmics analisi.

```
aws omics accept-share \  
  ----share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Output:

```
{  
  "status": "ACTIVATING"  
}
```

Per ulteriori informazioni, consulta [Condivisione tra account](#) nella Guida per l'AWS HealthOmics utente.

- Per API i dettagli, vedere [AcceptShare](#) in AWS CLI Command Reference.

## batch-delete-read-set

Il seguente esempio di codice mostra come utilizzare `batch-delete-read-set`.

## AWS CLI

Per eliminare più set di lettura

L'`batch-delete-read-set` seguente elimina due set di lettura.

```
aws omics batch-delete-read-set \  
  --sequence-store-id 1234567890 \  
  --ids 1234567890 0123456789
```

Se si verifica un errore durante l'eliminazione di uno dei set di lettura specificati, il servizio restituisce un elenco di errori.

```
{  
  "errors": [  
    {  
      "code": "",  
      "id": "0123456789",  
      "message": "The specified readset does not exist."  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [BatchDeleteReadSet](#) Reference.

## `cancel-annotation-import-job`

Il seguente esempio di codice mostra come utilizzare `cancel-annotation-import-job`.

## AWS CLI

Per annullare un processo di importazione di annotazioni

L'`cancel-annotation-import-job` seguente annulla un processo di importazione di annotazioni con ID. `04f57618-xmpl-4fd0-9349-e5a85aefb997`

```
aws omics cancel-annotation-import-job \  
  --job-id 04f57618-xmpl-4fd0-9349-e5a85aefb997
```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CancelAnnotationImportJobReference](#).

## cancel-run

Il seguente esempio di codice mostra come utilizzare `cancel-run`.

### AWS CLI

Per annullare una corsa

L'`cancel-run` esempio seguente annulla un'esecuzione con ID. 1234567

```
aws omics cancel-run \  
  --id 1234567
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [CancelRunReference](#) AWS CLI .

## cancel-variant-import-job

Il seguente esempio di codice mostra come utilizzare `cancel-variant-import-job`.

### AWS CLI

Per annullare un processo di importazione di varianti

L'`cancel-variant-import-job` esempio seguente annulla un processo di importazione di varianti con ID. 69cb65d6-xmpl-4a4a-9025-4565794b684e

```
aws omics cancel-variant-import-job \  
  --job-id 69cb65d6-xmpl-4a4a-9025-4565794b684e
```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CancelVariantImportJobReference](#).

## complete-multipart-read-set-upload

Il seguente esempio di codice mostra come utilizzare `complete-multipart-read-set-upload`.



## AWS CLI

Per concludere un caricamento in più parti dopo aver caricato tutti i componenti.

L'`complete-multipart-read-set-upload` seguente conclude un caricamento in più parti in un archivio di sequenze una volta caricati tutti i componenti.

```
aws omics complete-multipart-read-set-upload \  
  --sequence-store-id 0123456789 \  
  --upload-id 1122334455 \  
  --parts '[{"checksum":"gaCBQMe+rpCFZxLpoP6gydBoXaKKDA/  
Vobh5zBDb4W4=", "partNumber":1, "partSource":"SOURCE1"}]'
```

Output:

```
{  
  "readSetId": "0000000001"  
  "readSetId": "0000000002"  
  "readSetId": "0000000003"  
}
```

Per ulteriori informazioni, consultate [Caricamento diretto su un archivio di sequenze nella Guida](#) per l'AWS HealthOmics utente.

- Per API i dettagli, consultate [CompleteMultipartReadSetUpload AWS CLI Command Reference](#).

## `create-annotation-store-version`

Il seguente esempio di codice mostra come utilizzare `create-annotation-store-version`.

## AWS CLI

Per creare una nuova versione di un archivio di annotazioni

L'`create-annotation-store-version` seguente crea una nuova versione di un archivio di annotazioni.

```
aws omics create-annotation-store-version \  
  --name my_annotation_store \  
  --version-name my_version
```

**Output:**

```
{
  "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"
}
```

Per ulteriori informazioni, vedere [Creazione di nuove versioni di archivi di annotazioni nella Guida](#) per l'AWS HealthOmics utente.

- Per API i dettagli, consulta [CreateAnnotationStoreVersion AWS CLI Command Reference](#).

**create-annotation-store**

Il seguente esempio di codice mostra come utilizzare `create-annotation-store`.

**AWS CLI**

Esempio 1: creare un archivio di VCF annotazioni

L'esempio seguente crea un archivio di annotazioni VCF in formato.

```
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
```

**Output:**

```
{
  "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"
  }

```

## Esempio 2: creare un archivio di TSV annotazioni

L'create-annotation-store esempio seguente crea un archivio di annotazioni TSV in formato.

```

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 configura le opzioni di formato per le annotazioni.

```

{
  "tsvStoreOptions": {
    "annotationType": "CHR_START_END_ZERO_BASE",
    "formatToHeader": {
      "CHR": "chromosome",
      "START": "start",
      "END": "end"
    },
    "schema": [
      {
        "chromosome": "STRING"
      },
      {
        "start": "LONG"
      },
      {
        "end": "LONG"
      }
    ]
  }
}

```

```
        "name": "STRING"
      }
    ]
  }
}
```

### Output:

```
{
  "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"
        }
      ]
    }
  }
}
```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateAnnotationStoreReference](#).

## create-multipart-read-set-upload

Il seguente esempio di codice mostra come utilizzare `create-multipart-read-set-upload`.

### AWS CLI

Per iniziare un caricamento di un set di lettura in più parti.

L'esempio seguente avvia il caricamento di un set di lettura in più parti.

```
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"
```

### Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, vedete [Caricamento diretto su un archivio di sequenze nella Guida](#) per l'AWS HealthOmics utente.

- Per API i dettagli, consultate [CreateMultipartReadSetUpload AWS CLI Command Reference](#).

## create-reference-store

Il seguente esempio di codice mostra come utilizzare `create-reference-store`.

### AWS CLI

Per creare un archivio di riferimento

L'esempio seguente crea un archivio di riferimento `my-ref-store`.

```
aws omics create-reference-store \  
  --name my-ref-store
```

Output:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",  
  "creationTime": "2022-11-22T22:13:25.947Z",  
  "id": "1234567890",  
  "name": "my-ref-store"  
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateReferenceStore](#)Reference.

## create-run-group

Il seguente esempio di codice mostra come utilizzare `create-run-group`.

### AWS CLI

Per creare un gruppo di corsa

L'esempio seguente crea un gruppo di esecuzioni denominato `cram-converter`.

```
aws omics create-run-group \  
  --name cram-converter \  
  --max-cpus 20 \  
  --max-duration 600
```

**Output:**

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
  "id": "1234567",
  "tags": {}
}
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [CreateRunGroup](#)Reference AWS CLI .

**create-sequence-store**

Il seguente esempio di codice mostra come utilizzare `create-sequence-store`.

**AWS CLI**

Per creare un archivio di sequenze

L'esempio seguente crea un archivio di sequenze.

```
aws omics create-sequence-store \
  --name my-seq-store
```

**Output:**

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",
  "creationTime": "2022-11-23T01:24:33.629Z",
  "id": "1234567890",
  "name": "my-seq-store"
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateSequenceStore](#)Reference.

**create-share**

Il seguente esempio di codice mostra come utilizzare `create-share`.

## AWS CLI

Per creare una condivisione di un archivio di HealthOmics analisi

L'`create-share` seguente mostra come creare una condivisione di un negozio di HealthOmics analisi che possa essere accettata da un abbonato esterno all'account.

```
aws omics create-share \  
  --resource-arn "arn:aws:omics:us-west-2:555555555555:variantStore/  
omics_dev_var_store" \  
  --principal-subscriber "123456789012" \  
  --name "my_Share-123"
```

Output:

```
{  
  "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",  
  "name": "my_Share-123",  
  "status": "PENDING"  
}
```

Per ulteriori informazioni, consulta [Condivisione tra account](#) nella Guida per l'AWS HealthOmics utente.

- Per API i dettagli, consulta AWS CLI Command [CreateShare](#)Reference.

## `create-variant-store`

Il seguente esempio di codice mostra come utilizzare `create-variant-store`.

## AWS CLI

Per creare un negozio di varianti

L'`create-variant-store` seguente crea un archivio di varianti denominato `my_var_store`.

```
aws omics create-variant-store \  
  --name my_var_store \  
  --reference referenceArn=arn:aws:omics:us-  
west-2:123456789012:referenceStore/1234567890/reference/1234567890
```



**Output:**

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateVariantStoreReference](#).

**create-workflow**

Il seguente esempio di codice mostra come utilizzare `create-workflow`.

**AWS CLI**

Per creare un flusso di lavoro

L'`create-workflow` esempio seguente crea un WDL flusso di lavoro.

```
aws omics create-workflow \
  --name cram-converter \
  --engine WDL \
  --definition-zip fileb://workflow-crambam.zip \
  --parameter-template file://workflow-params.json
```

`workflow-crambam.zip` è un ZIP archivio contenente una definizione del flusso di lavoro.

`workflow-params.json` definisce i parametri di runtime per il flusso di lavoro.

```
{
  "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
  }
}
```

Output:

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
  "id": "1234567",
  "status": "CREATING",
  "tags": {}
}
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [CreateWorkflow](#)Reference AWS CLI .

## **delete-annotation-store-versions**

Il seguente esempio di codice mostra come utilizzare `delete-annotation-store-versions`.

AWS CLI

Per eliminare una versione dell'archivio delle annotazioni

L'`delete-annotation-store-versions`esempio seguente elimina una versione dell'archivio delle annotazioni.

```
aws omics delete-annotation-store-versions \
```

```
--name my_annotation_store \  
--versions my_version
```

Output:

```
{  
  "errors": []  
}
```

Per ulteriori informazioni, vedere [Creazione di nuove versioni degli archivi di annotazioni](#) nella Guida per l'AWS HealthOmics utente.

- Per API i dettagli, consulta [DeleteAnnotationStoreVersions AWS CLI Command Reference](#).

## delete-annotation-store

Il seguente esempio di codice mostra come utilizzare `delete-annotation-store`.

AWS CLI

Per eliminare un archivio di annotazioni

L'esempio seguente elimina un archivio di annotazioni denominato `my_vcf_store`

```
aws omics delete-annotation-store \  
  --name my_vcf_store
```

Output:

```
{  
  "status": "DELETING"  
}
```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteAnnotationStore](#) Reference.

## delete-reference-store

Il seguente esempio di codice mostra come utilizzare `delete-reference-store`.

## AWS CLI

Per eliminare un archivio di riferimento

L'`delete-reference-store` seguente elimina un archivio di riferimenti con ID. 1234567890

```
aws omics delete-reference-store \  
  --id 1234567890
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteReferenceStore](#) Reference.

## `delete-reference`

Il seguente esempio di codice mostra come utilizzare `delete-reference`.

### AWS CLI

Per eliminare un riferimento

L'`delete-reference` seguente elimina un riferimento.

```
aws omics delete-reference \  
  --reference-store-id 1234567890 \  
  --id 1234567890
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteReference](#) Reference.

## `delete-run-group`

Il seguente esempio di codice mostra come utilizzare `delete-run-group`.

### AWS CLI

Per eliminare un gruppo di esecuzione

L'`delete-run-group` seguente elimina un gruppo di esecuzione con ID. 1234567

```
aws omics delete-run-group \  
--id 1234567
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [DeleteRunGroup](#)Reference AWS CLI .

## delete-run

Il seguente esempio di codice mostra come utilizzare `delete-run`.

### AWS CLI

Per eliminare un flusso di lavoro, esegui

L'`delete-run`esempio seguente elimina un'esecuzione con ID. 1234567

```
aws omics delete-run \  
--id 1234567
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [DeleteRun](#)Reference AWS CLI .

## delete-sequence-store

Il seguente esempio di codice mostra come utilizzare `delete-sequence-store`.

### AWS CLI

Per eliminare un archivio di sequenze

L'`delete-sequence-store`esempio seguente elimina un archivio di sequenze con ID. 1234567890

```
aws omics delete-sequence-store \  
--id 1234567890
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteSequenceStore](#)Reference.

## delete-share

Il seguente esempio di codice mostra come utilizzare `delete-share`.

### AWS CLI

Per eliminare una condivisione di dati di HealthOmics analisi

L'esempio seguente elimina una condivisione di dati di analisi tra account.

```
aws omics delete-share \  
  --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Output:

```
{  
  "status": "DELETING"  
}
```

Per ulteriori informazioni, consulta [Condivisione tra account](#) nella Guida per l'AWS HealthOmics utente.

- Per API i dettagli, vedere [DeleteShare](#) in AWS CLI Command Reference.

## delete-variant-store

Il seguente esempio di codice mostra come utilizzare `delete-variant-store`.

### AWS CLI

Per eliminare un archivio di varianti

L'esempio seguente elimina un archivio di varianti denominato `my_var_store`.

```
aws omics delete-variant-store \  
  --name my_var_store
```

Output:

```
{
```

```
"status": "DELETING"  
}
```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteVariantStore](#)Reference.

## **delete-workflow**

Il seguente esempio di codice mostra come utilizzare `delete-workflow`.

### AWS CLI

Per eliminare un flusso di lavoro

L'`delete-workflow`esempio seguente elimina un flusso di lavoro con ID. 1234567

```
aws omics delete-workflow \  
  --id 1234567
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [DeleteWorkflow](#)Reference AWS CLI .

## **get-annotation-import-job**

Il seguente esempio di codice mostra come utilizzare `get-annotation-import-job`.

### AWS CLI

Per visualizzare un processo di importazione di annotazioni

L'`get-annotation-import-job`esempio seguente ottiene dettagli su un processo di importazione di annotazioni.

```
aws omics get-annotation-import-job \  
  --job-id 984162c7-xmpl-4d23-ab47-286f7950bfbf
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetAnnotationImportJob](#) Reference.

## get-annotation-store-version

Il seguente esempio di codice mostra come utilizzare `get-annotation-store-version`.

### AWS CLI

Per recuperare i metadati per una versione dell'archivio di annotazioni

L'`get-annotation-store-version` esempio seguente recupera i metadati per la versione dell'annotation store richiesta.

```

aws omics get-annotation-store-version \
  --name my_annotation_store \
  --version-name my_version

```

Output:

```

{
  "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
}
```

Per ulteriori informazioni, consultate [Creazione di nuove versioni degli archivi di annotazioni](#) nella Guida per l'utente.AWS HealthOmics

- Per API i dettagli, consulta [GetAnnotationStoreVersion AWS CLI Command Reference](#).

## get-annotation-store

Il seguente esempio di codice mostra come utilizzare `get-annotation-store`.

### AWS CLI

Per visualizzare un archivio di annotazioni

L'`get-annotation-store`esempio seguente ottiene dettagli su un archivio di annotazioni denominato `my_ann_store`

```
aws omics get-annotation-store \
  --name my_ann_store
```

Output:

```
{
  "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": {}
}
```

```
}
```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetAnnotationStoreReference](#).

## get-read-set-activation-job

Il seguente esempio di codice mostra come utilizzare `get-read-set-activation-job`.

### AWS CLI

Per visualizzare un processo di attivazione del set di lettura

L'`get-read-set-activation-job`esempio seguente fornisce dettagli su un processo di attivazione del set di lettura.

```
aws omics get-read-set-activation-job \  
  --sequence-store-id 1234567890 \  
  --id 1234567890
```

Output:

```
{  
  "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."  
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetReadSetActivationJobReference](#).

## get-read-set-export-job

Il seguente esempio di codice mostra come utilizzare `get-read-set-export-job`.

### AWS CLI

Per visualizzare un processo di esportazione con set di lettura

L'`get-read-set-export-job` seguente fornisce dettagli su un processo di esportazione di set di lettura.

```
aws omics get-read-set-export-job \  
  --sequence-store-id 1234567890 \  
  --id 1234567890
```

Output:

```
{  
  "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."  
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetReadSetExportJob](#) Reference.

## get-read-set-import-job

Il seguente esempio di codice mostra come utilizzare `get-read-set-import-job`.

### AWS CLI

Per visualizzare un processo di importazione di set di lettura

L'`get-read-set-import-job` seguente fornisce dettagli su un processo di importazione di set di lettura.

```
aws omics get-read-set-import-job \  
  --sequence-store-id 1234567890 \  
  --id 1234567890
```

### Output:

```
{  
  "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."
}

```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetReadSetImportJobReference](#).

## get-read-set-metadata

Il seguente esempio di codice mostra come utilizzare `get-read-set-metadata`.

### AWS CLI

Per visualizzare un set di lettura

L'get-read-set-metadata esempio seguente ottiene dettagli sui file di un set di lettura.

```
aws omics get-read-set-metadata \  
  --sequence-store-id 1234567890 \  
  --id 1234567890
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetReadSetMetadata](#) Reference.

## get-read-set

Il seguente esempio di codice mostra come utilizzare `get-read-set`.

### AWS CLI

Per scaricare un set di lettura

L'`get-read-set`esempio seguente scarica la parte 3 di un set di lettura come `1234567890.3.bam`.

```
aws omics get-read-set \  
  --sequence-store-id 1234567890 \  
  --id 1234567890 \  
  --part-number 3 1234567890.3.bam
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetReadSet](#)Reference.

## get-reference-import-job

Il seguente esempio di codice mostra come utilizzare `get-reference-import-job`.

### AWS CLI

Per visualizzare un processo di importazione di riferimenti

L'`get-reference-import-job`esempio seguente ottiene dettagli su un processo di importazione di riferimenti.

```
aws omics get-reference-import-job \  
  --reference-store-id 1234567890 \  
  --id 1234567890
```

Output:

```
{  
  "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."
  }

```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetReferenceImportJobReference](#).

## get-reference-metadata

Il seguente esempio di codice mostra come utilizzare `get-reference-metadata`.

### AWS CLI

Per visualizzare un riferimento

L'`get-reference-metadata` esempio seguente fornisce dettagli su un riferimento.

```

aws omics get-reference-metadata \
  --reference-store-id 1234567890 \
  --id 1234567890

```

Output:

```

{
  "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"
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetReferenceMetadata](#)Reference.

## get-reference-store

Il seguente esempio di codice mostra come utilizzare `get-reference-store`.

### AWS CLI

Per visualizzare un negozio di riferimento

L'`get-reference-store`esempio seguente fornisce dettagli su un negozio di riferimento.

```
aws omics get-reference-store \
  --id 1234567890
```

Output:

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
  "creationTime": "2022-09-23T23:27:20.364Z",
  "id": "1234567890",
  "name": "my-rstore-0"
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetReferenceStore](#)Reference.

## get-reference

Il seguente esempio di codice mostra come utilizzare `get-reference`.

### AWS CLI

Per scaricare un riferimento al genoma

L'`get-reference` esempio seguente scarica la parte 1 di un genoma come `hg38.1.fa`

```
aws omics get-reference \  
  --reference-store-id 1234567890 \  
  --id 1234567890 \  
  --part-number 1 hg38.1.fa
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetReference](#) Reference.

## get-run-group

Il seguente esempio di codice mostra come utilizzare `get-run-group`.

### AWS CLI

Per visualizzare un gruppo di corsa

L'`get-run-group` esempio seguente fornisce dettagli su un gruppo di corsa.

```
aws omics get-run-group \  
  --id 1234567
```

Output:

```
{  
  "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": {}
```

```
}
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [GetRunGroup](#)Reference AWS CLI .

## get-run-task

Il seguente esempio di codice mostra come utilizzare `get-run-task`.

### AWS CLI

Per visualizzare un'attività

L'`get-run-task`esempio seguente fornisce dettagli su un'attività del flusso di lavoro.

```
aws omics get-run-task \  
  --id 1234567 \  
  --task-id 1234567
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [GetRunTask](#)Reference AWS CLI .

## get-run

Il seguente esempio di codice mostra come utilizzare `get-run`.

## AWS CLI

Per visualizzare un flusso di lavoro, esegui

L'get-run seguente fornisce dettagli sull'esecuzione di un flusso di lavoro.

```
aws omics get-run \  
  --id 1234567
```

Output:

```
{  
  "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",  
  "creationTime": "2022-11-30T22:58:22.615865Z",  
  "digest":  
    "sha256:c54bxmpl742dcc26f7fa1f10e37550ddd8f251f418277c0a58e895b801ed28cf",  
  "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:3884c62eb0e53fa92459ed9bfff133ae6",  
    "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"
}
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [GetRun](#)Reference AWS CLI .

## get-sequence-store

Il seguente esempio di codice mostra come utilizzare `get-sequence-store`.

### AWS CLI

Per visualizzare un archivio di sequenze

L'`get-sequence-store`esempio seguente ottiene dettagli su un archivio di sequenze con ID1234567890.

```
aws omics get-sequence-store \
  --id 1234567890
```

Output:

```
{
  "arn": "arn:aws:omics:us-east-1:123456789012:sequenceStore/1234567890",
  "creationTime": "2022-11-23T19:55:48.376Z",
  "id": "1234567890",
  "name": "my-seq-store"
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetSequenceStore](#)Reference.

## get-share

Il seguente esempio di codice mostra come utilizzare `get-share`.

### AWS CLI

Per recuperare i metadati relativi a una condivisione di dati di analisi HealthOmics

L'get-shareesempio seguente recupera i metadati per una condivisione di dati di analisi tra account.

```
aws omics get-share \  
  --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Condivisione tra account](#) nella Guida per l'utente.AWS HealthOmics

- Per API i dettagli, vedere [GetShare](#)in AWS CLI Command Reference.

## get-variant-import-job

Il seguente esempio di codice mostra come utilizzareget-variant-import-job.

AWS CLI

Per visualizzare un processo di importazione di varianti

L'get-variant-import-jobesempio seguente fornisce dettagli su un processo di importazione di varianti.

```
aws omics get-variant-import-job \  
  --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetVariantImportJob](#) Reference.

## get-variant-store

Il seguente esempio di codice mostra come utilizzare `get-variant-store`.

### AWS CLI

Per visualizzare un negozio di varianti

L'`get-variant-store` esempio seguente fornisce dettagli su un negozio di varianti.

```
aws omics get-variant-store \
  --name my_var_store
```

Output:

```
{
  "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,
  "tags": {},
  "updateTime": "2022-11-23T22:09:24.931711Z"
}

```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetVariantStoreReference](#).

## get-workflow

Il seguente esempio di codice mostra come utilizzare `get-workflow`.

### AWS CLI

Per visualizzare un flusso di lavoro

L'`get-workflow` esempio seguente fornisce dettagli su un flusso di lavoro con ID1234567.

```

aws omics get-workflow \
  --id 1234567

```

Output:

```

{
  "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"
}

```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [GetWorkflow](#)Reference AWS CLI .

## list-annotation-import-jobs

Il seguente esempio di codice mostra come utilizzare `list-annotation-import-jobs`.

AWS CLI

Per ottenere un elenco dei lavori di importazione delle annotazioni

Di seguito `list-annotation-import-jobs` viene visualizzato un elenco dei lavori di importazione delle annotazioni.

```
aws omics list-annotation-import-jobs
```

Output:

```
{
```

```

"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"
  }
]
}

```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListAnnotationImportJobsReference](#).

## list-annotation-store-versions

Il seguente esempio di codice mostra come utilizzare `list-annotation-store-versions`.

### AWS CLI

Per elencare tutte le versioni di un archivio di annotazioni.

L'`list-annotation-store-versions`esempio seguente elenca tutte le versioni esistenti di un archivio di annotazioni.

```

aws omics list-annotation-store-versions \
  --name my_annotation_store

```

Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, vedere [Creazione di nuove versioni di archivi di annotazioni nella Guida per l'AWS HealthOmics utente](#).

- Per API i dettagli, consulta [ListAnnotationStoreVersions AWS CLI Command Reference](#).

## list-annotation-stores

Il seguente esempio di codice mostra come utilizzare `list-annotation-stores`.

### AWS CLI

Per ottenere un elenco di archivi di annotazioni

L'`list-annotation-stores` seguente ottiene un elenco di archivi di annotazioni.

```
aws omics list-annotation-stores
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListAnnotationStores](#)Reference.

## list-multipart-read-set-uploads

Il seguente esempio di codice mostra come utilizzare `list-multipart-read-set-uploads`.

AWS CLI

Per elencare tutti i caricamenti di set di lettura in più parti e i relativi stati.

L'esempio seguente elenca tutti i caricamenti di set di lettura in più parti e i relativi stati.

```
aws omics list-multipart-read-set-uploads \
```

```
--sequence-store-id 0123456789
```

## Output:

```
{
  "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"
      }
    ]
}
```

```
    }  
  ]  
}
```

Per ulteriori informazioni, consultate [Caricamento diretto su un archivio di sequenze nella Guida](#) per l'AWS HealthOmics utente.

- Per API i dettagli, consultate [ListMultipartReadSetUploads AWS CLI](#) Command Reference.

## list-read-set-activation-jobs

Il seguente esempio di codice mostra come utilizzare `list-read-set-activation-jobs`.

### AWS CLI

Per ottenere un elenco dei processi di attivazione dei set di lettura

L'`list-read-set-activation-jobs` esempio seguente ottiene un elenco di processi di attivazione per un archivio di sequenze con id `1234567890`.

```
aws omics list-read-set-activation-jobs \  
  --sequence-store-id 1234567890
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListReadSetActivationJobs](#)Reference.

## list-read-set-export-jobs

Il seguente esempio di codice mostra come utilizzare `list-read-set-export-jobs`.

### AWS CLI

To ottiene un elenco di lavori di esportazione con set di lettura

L'`list-read-set-export-jobs`esempio seguente ottiene un elenco di processi di esportazione per un archivio di sequenze con id1234567890.

```
aws omics list-read-set-export-jobs \  
  --sequence-store-id 1234567890
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListReadSetExportJobs](#) Reference.

## list-read-set-import-jobs

Il seguente esempio di codice mostra come utilizzare `list-read-set-import-jobs`.

### AWS CLI

Per ottenere un elenco dei lavori di importazione dei set di lettura

L'`list-read-set-import-jobs` seguente ottiene un elenco di lavori di importazione per un archivio di sequenze con id `1234567890`.

```
aws omics list-read-set-import-jobs \  
  --sequence-store-id 1234567890
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.



- Per API i dettagli, consulta AWS CLI Command [ListReadSetImportJobs](#)Reference.

## list-read-set-upload-parts

Il seguente esempio di codice mostra come utilizzare `list-read-set-upload-parts`.

### AWS CLI

Per elencare tutte le parti in un caricamento multiparte richiesto per un archivio di sequenze.

L'`list-read-set-upload-parts`esempio seguente elenca tutte le parti in un caricamento multiparte richiesto per un archivio di sequenze.

```
aws omics list-read-set-upload-parts \
  --sequence-store-id 0123456789 \
  --upload-id 1122334455 \
  --part-source SOURCE1
```

### Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consultate [Caricamento diretto su un archivio di sequenze nella Guida per l'AWS HealthOmics utente](#).

- Per API i dettagli, consultate [ListReadSetUploadParts AWS CLI Command Reference](#).

## list-read-sets

Il seguente esempio di codice mostra come utilizzare `list-read-sets`.

### AWS CLI

Per ottenere un elenco di set di lettura

L'`list-read-sets` seguente ottiene un elenco di set di lettura per un archivio di sequenze con id `1234567890`.

```
aws omics list-read-sets \  
  --sequence-store-id 1234567890
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListReadSets](#)Reference.

## list-reference-import-jobs

Il seguente esempio di codice mostra come utilizzare `list-reference-import-jobs`.

### AWS CLI

Per ottenere un elenco di lavori di importazione di riferimento

L'`list-reference-import-jobs`esempio seguente ottiene un elenco di lavori di importazione di riferimento per un archivio di riferimenti con id1234567890.

```
aws omics list-reference-import-jobs \  
--reference-store-id 1234567890
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListReferenceImportJobs](#)Reference.

## list-reference-stores

Il seguente esempio di codice mostra come utilizzare `list-reference-stores`.

### AWS CLI

Per ottenere un elenco di negozi di riferimento

L'`list-reference-stores`esempio seguente ottiene un elenco di negozi di riferimento.

```
aws omics list-reference-stores
```

Output:

```
{
  "referenceStores": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
      "creationTime": "2022-11-22T22:13:25.947Z",
      "id": "1234567890",
      "name": "my-ref-store"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListReferenceStores](#)Reference.

## list-references

Il seguente esempio di codice mostra come utilizzare `list-references`.

### AWS CLI

Per ottenere un elenco di riferimenti

L'`list-references`esempio seguente ottiene un elenco di riferimenti genomici per un archivio di riferimenti con id1234567890.

```
aws omics list-references \  
--reference-store-id 1234567890
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListReferences](#)Reference.

## list-run-groups

Il seguente esempio di codice mostra come utilizzare `list-run-groups`.

AWS CLI

Per ottenere un elenco di gruppi di esecuzione

L'`list-run-groups`esempio seguente ottiene un elenco di gruppi di esecuzione.

```
aws omics list-run-groups
```

Output:

```
{  
  "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"
}
]
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [ListRunGroups](#)Reference AWS CLI .

## list-run-tasks

Il seguente esempio di codice mostra come utilizzare `list-run-tasks`.

### AWS CLI

Per ottenere un elenco di attività

L'`list-run-tasks`esempio seguente ottiene un elenco di attività per l'esecuzione di un flusso di lavoro.

```
aws omics list-run-tasks \
  --id 1234567
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [ListRunTasks](#)Reference AWS CLI .

## list-runs

Il seguente esempio di codice mostra come utilizzare `list-runs`.

### AWS CLI

Per ottenere un elenco delle esecuzioni del flusso di lavoro

L'`list-runs`esempio seguente ottiene un elenco di esecuzioni del flusso di lavoro.

```
aws omics list-runs
```

Output:

```
{
  "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"
  }
]
}
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [ListRuns](#)Reference AWS CLI .

## list-sequence-stores

Il seguente esempio di codice mostra come utilizzare `list-sequence-stores`.

### AWS CLI

Per ottenere un elenco di archivi di sequenze

L'`list-sequence-stores`esempio seguente ottiene un elenco di archivi di sequenze.

```
aws omics list-sequence-stores
```

Output:



```
{
  "sequenceStores": [
    {
      "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",
      "creationTime": "2022-11-23T01:24:33.629Z",
      "id": "1234567890",
      "name": "my-seq-store"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListSequenceStores](#) Reference.

## list-shares

Il seguente esempio di codice mostra come utilizzare `list-shares`.

### AWS CLI

Per elencare le condivisioni disponibili di un dato di HealthOmics analisi

L'`list-shares` esempio seguente elenca tutte le condivisioni che sono state create per il proprietario di una risorsa.

```
aws omics list-shares \
  --resource-owner SELF
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Condivisione tra account nella Guida](#) per l'utente.AWS HealthOmics

- Per API i dettagli, vedere [ListShares](#)in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per ottenere un elenco di tag

L'`list-tags-for-resource`esempio seguente ottiene un elenco di tag per un flusso di lavoro con id1234567.

```
aws omics list-tags-for-resource \
  --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567
```

Output:

```
{
  "tags": {
    "department": "analytics"
  }
}
```

Per ulteriori informazioni, consulta [Tagging resources in Amazon Omics nella Amazon Omics Developer Guide](#).

- Per API i dettagli, consulta Command [ListTagsForResource](#)Reference AWS CLI .

## list-variant-import-jobs

Il seguente esempio di codice mostra come utilizzare `list-variant-import-jobs`.

### AWS CLI

Per ottenere un elenco di processi di importazione di varianti

L'`list-variant-import-jobs`esempio seguente ottiene un elenco di processi di importazione di varianti.

```
aws omics list-variant-import-jobs
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListVariantImportJobs](#) Reference.

## list-variant-stores

Il seguente esempio di codice mostra come utilizzare `list-variant-stores`.

### AWS CLI

Per ottenere un elenco di negozi di varianti

L'`list-variant-stores` seguente ottiene un elenco di negozi di varianti.

```
aws omics list-variant-stores
```

Output:

```

{
  "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"
}
]
```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListVariantStores](#) Reference.

## list-workflows

Il seguente esempio di codice mostra come utilizzare `list-workflows`.

### AWS CLI

Per ottenere un elenco di flussi di lavoro

L'`list-workflows` esempio seguente ottiene un elenco di flussi di lavoro.

```
aws omics list-workflows
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [ListWorkflowsReference](#) AWS CLI .

## start-annotation-import-job

Il seguente esempio di codice mostra come utilizzare `start-annotation-import-job`.

AWS CLI

Per importare annotazioni

L'`start-annotation-import-job`esempio seguente importa annotazioni da 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

```

Output:

```

{
  "jobId": "984162c7-xmpl-4d23-ab47-286f7950bfbf"
}

```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [StartAnnotationImportJobReference](#).



```
--destination s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/
```

Output:

```
{
  "creationTime": "2022-12-06T22:37:18.612Z",
  "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
  "id": "1234567890",
  "sequenceStoreId": "1234567890",
  "status": "SUBMITTED"
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [StartReadSetExportJob](#) Reference.

## start-read-set-import-job

Il seguente esempio di codice mostra come utilizzare `start-read-set-import-job`.

AWS CLI

Per importare un set di lettura

L'`start-read-set-import-job` esempio seguente importa un set di lettura.

```
aws omics start-read-set-import-job \
  --sequence-store-id 1234567890 \
  --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ \
  --sources file://readset-sources.json
```

`readset-sources.json` è un documento con il seguente contenuto. JSON

```
[
  {
    "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"
  }
]
```

Output:

```
{
  "creationTime": "2022-11-23T01:36:38.158Z",
  "id": "1234567890",
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ",
  "sequenceStoreId": "1234567890",
  "status": "SUBMITTED"
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [StartReadSetImportJobReference](#).

## start-reference-import-job

Il seguente esempio di codice mostra come utilizzare `start-reference-import-job`.

### AWS CLI

Per importare un genoma di riferimento

L'esempio seguente importa un genoma di riferimento da 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
```

Output:

```
{
  "creationTime": "2022-11-22T22:25:41.124Z",
  "id": "1234567890",
  "referenceStoreId": "1234567890",
  "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ",
  "status": "SUBMITTED"
}
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [StartReferenceImportJob](#) Reference.

## start-run

Il seguente esempio di codice mostra come utilizzare `start-run`.

### AWS CLI

Per eseguire un flusso di lavoro

L'`start-run` esempio seguente esegue un flusso di lavoro con ID1234567.

```
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` è un documento con il seguente contenuto. JSON

```
{
  "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"
}
```

Output:

```
{
  "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
  "id": "1234567",
  "status": "PENDING",
  "tags": {}
}
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

Per caricare file sorgente da Amazon Omics

Puoi anche caricare file sorgente dallo storage Amazon Omics, utilizzando servizi URIs specifici. Il seguente file workflow-inputs.json di esempio utilizza Amazon URIs Omics per i set di lettura e le fonti genomiche di riferimento.

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [StartRun](#)Reference AWS CLI .

## start-variant-import-job

Il seguente esempio di codice mostra come utilizzare `start-variant-import-job`.

## AWS CLI

Per importare un file di variante

L'`start-variant-import-job` seguente importa un file di variante di VCF formato.

```
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-  
W801XMPL7QZ \  
  --items source=s3://omics-artifacts-01d6xmpl4e72dd32/  
Homo_sapiens_assembly38.known_indels.vcf.gz
```

Output:

```
{  
  "jobId": "edd7b8ce-xmpl-47e2-bc99-258cac95a508"  
}
```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [StartVariantImportJob](#) Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

## AWS CLI

Per etichettare una risorsa

L'`tag-resource` seguente aggiunge un `department` tag a un flusso di lavoro con `id1234567`.

```
aws omics tag-resource \  
  --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \  
  --tags department=analytics
```

Per ulteriori informazioni, consulta [Tagging resources in Amazon Omics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [TagResource](#)Reference AWS CLI .

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da una risorsa

L'`untag-resource`esempio seguente rimuove il `department` tag da un flusso di lavoro.

```
aws omics untag-resource \  
  --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \  
  --tag-keys department
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UntagResource](#)Reference.

## update-annotation-store

Il seguente esempio di codice mostra come utilizzare `update-annotation-store`.

### AWS CLI

Per aggiornare un archivio di annotazioni

L'`update-annotation-store`esempio seguente aggiorna la descrizione di un archivio di annotazioni denominato `my_vcf_store`

```
aws omics update-annotation-store \  
  --name my_vcf_store \  
  --description "VCF annotation store"
```

Output:

```
{  
  "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"
}
```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateAnnotationStoreReference](#).

## update-run-group

Il seguente esempio di codice mostra come utilizzare `update-run-group`.

### AWS CLI

Per aggiornare un gruppo di esecuzione

L'`update-run-group` seguente aggiorna le impostazioni di un gruppo di esecuzione con id1234567.

```
aws omics update-run-group \
  --id 1234567 \
  --max-cpus 10
```

Output:

```
{
  "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": {}
}
```

Per ulteriori informazioni, consulta [Omics Workflows](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta Command [UpdateRunGroup](#)Reference AWS CLI .

## update-variant-store

Il seguente esempio di codice mostra come utilizzare `update-variant-store`.

### AWS CLI

Per aggiornare un archivio di varianti

L'`update-variant-store`esempio seguente aggiorna la descrizione di un archivio di varianti denominato `my_var_store`.

```
aws omics update-variant-store \  
  --name my_var_store \  
  --description "variant store"
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Omics Analytics](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateVariantStore](#)Reference.

## update-workflow

Il seguente esempio di codice mostra come utilizzare `update-workflow`.

## AWS CLI

Per aggiornare un flusso di lavoro

L'update-workflow esempio seguente aggiorna la descrizione di un flusso di lavoro con ID1234567.

```
aws omics update-workflow \  
  --id 1234567 \  
  --description "copy workflow"
```

Per ulteriori informazioni, consulta [Omics Storage](#) nella Amazon Omics Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateWorkflow](#) Reference.

## upload-read-set-part

Il seguente esempio di codice mostra come utilizzare upload-read-set-part.

### AWS CLI

Per caricare una parte del set di lettura.

L'upload-read-set-part esempio seguente carica una parte specificata di un set di lettura.

```
aws omics upload-read-set-part \  
  --sequence-store-id 0123456789 \  
  --upload-id 1122334455 \  
  --part-source SOURCE1 \  
  --part-number 1 \  
  --payload /path/to/file/read_1_part_1.fastq.gz
```

Output:

```
{  
  "checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"  
}
```

Per ulteriori informazioni, vedete [Caricamento diretto su un archivio di sequenze nella Guida per l'AWS HealthOmics utente](#).

- Per API i dettagli, consultate [UploadReadSetPart AWS CLI](#) Command Reference.



## IAM esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with IAM.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **add-client-id-to-open-id-connect-provider**

Il seguente esempio di codice mostra come utilizzare `add-client-id-to-open-id-connect-provider`.

AWS CLI

Per aggiungere un ID client (audience) a un provider Open-ID Connect (OIDC)

Il `add-client-id-to-open-id-connect-provider` comando seguente aggiunge l'ID client `my-application-ID` al OIDC provider denominato `server.example.com`.

```
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
```

Questo comando non produce alcun output.

Per creare un OIDC provider, utilizzare il `create-open-id-connect-provider` comando.

Per ulteriori informazioni, consulta [Creazione di provider di identità OpenID Connect \(OIDC\) nella Guida](#) per l'AWS IAM utente.

- Per API i dettagli, consultate [AddClientIdToOpenIdConnectProvider AWS CLI Command Reference](#).

## add-role-to-instance-profile

Il seguente esempio di codice mostra come utilizzare `add-role-to-instance-profile`.

### AWS CLI

Per aggiungere un ruolo a un profilo di istanza

Il `add-role-to-instance-profile` comando seguente aggiunge il ruolo denominato `S3Access` al profilo di istanza denominato `Webserver`.

```
aws iam add-role-to-instance-profile \  
  --role-name S3Access \  
  --instance-profile-name Webserver
```

Questo comando non produce alcun output.

Per creare un profilo di istanza, utilizzate il `create-instance-profile` comando.

Per ulteriori informazioni, consulta [Usare un IAM ruolo per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella Guida](#) per l'AWS IAM utente.

- Per API i dettagli, consulta AWS CLI Command [AddRoleToInstanceProfile Reference](#).

## add-user-to-group

Il seguente esempio di codice mostra come utilizzare `add-user-to-group`.

### AWS CLI

Per aggiungere un utente a un IAM gruppo

Il `add-user-to-group` comando seguente aggiunge un IAM utente denominato `Bob` al IAM gruppo denominato `Admins`.

```
aws iam add-user-to-group \  
  --user-name Bob \  
  --group-name Admins
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Aggiungere e rimuovere utenti in un gruppo di IAM utenti](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [AddUserToGroup](#) in AWS CLI Command Reference.

## **attach-group-policy**

Il seguente esempio di codice mostra come utilizzare `attach-group-policy`.

### AWS CLI

Per allegare una politica gestita a un IAM gruppo

Il `attach-group-policy` comando seguente collega la politica AWS gestita denominata `ReadOnlyAccess` al IAM gruppo denominato `Finance`.

```
aws iam attach-group-policy \  
  --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \  
  --group-name Finance
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Politiche gestite e politiche in linea nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [AttachGroupPolicy](#) in AWS CLI Command Reference.

## **attach-role-policy**

Il seguente esempio di codice mostra come utilizzare `attach-role-policy`.

### AWS CLI

Per allegare una politica gestita a un IAM ruolo

Il `attach-role-policy` comando seguente collega la politica AWS gestita denominata `ReadOnlyAccess` al IAM ruolo denominato `ReadOnlyRole`.

```
aws iam attach-role-policy \  
  --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \  
  --role-name ReadOnlyRole
```

```
--role-name ReadOnlyRole
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Politiche gestite e politiche in linea nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [AttachRolePolicy](#) in AWS CLI Command Reference.

## attach-user-policy

Il seguente esempio di codice mostra come utilizzare `attach-user-policy`.

### AWS CLI

Per allegare una politica gestita a un IAM utente

Il `attach-user-policy` comando seguente collega la politica AWS gestita denominata `AdministratorAccess` all'IAMutente denominato `Alice`.

```
aws iam attach-user-policy \  
  --policy-arn arn:aws:iam::aws:policy/AdministratorAccess \  
  --user-name Alice
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Politiche gestite e politiche in linea nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [AttachUserPolicy](#) in AWS CLI Command Reference.

## change-password

Il seguente esempio di codice mostra come utilizzare `change-password`.

### AWS CLI

Per modificare la password IAM dell'utente

Per modificare la password IAM dell'utente, si consiglia di utilizzare il `--cli-input-json` parametro per passare un JSON file contenente la vecchia e la nuova password. Utilizzando questo metodo, è possibile utilizzare password complesse con caratteri non alfanumerici. Può

essere difficile utilizzare password con caratteri non alfanumerici quando le si passano come parametri della riga di comando. Per utilizzare il `--cli-input-json` parametro, iniziate a utilizzare il `change-password` comando con il `--generate-cli-skeleton` parametro, come nell'esempio seguente.

```
aws iam change-password \  
  --generate-cli-skeleton > change-password.json
```

Il comando precedente crea un JSON file chiamato `change-password.json` che potete utilizzare per inserire la vecchia e la nuova password. Ad esempio, il file potrebbe avere il seguente aspetto.

```
{  
  "OldPassword": "3s0K_;xh4~8XXI",  
  "NewPassword": "]35d/{pB9Fo9wJ"  
}
```

Quindi, per modificare la password, usa nuovamente il `change-password` comando, questa volta passando il `--cli-input-json` parametro per specificare il JSON file. Il `change-password` comando seguente utilizza il `--cli-input-json` parametro con un JSON file chiamato `change-password.json`.

```
aws iam change-password \  
  --cli-input-json file://change-password.json
```

Questo comando non produce alcun output.

Questo comando può essere richiamato solo dagli utenti IAM. Se questo comando viene chiamato utilizzando le credenziali AWS dell'account (root), restituisce un `InvalidUserType` errore.

Per ulteriori informazioni, vedere [Come un IAM utente modifica la propria password](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [ChangePassword](#) in AWS CLI Command Reference.

## create-access-key

Il seguente esempio di codice mostra come utilizzare `create-access-key`.

### AWS CLI

Per creare una chiave di accesso per un IAM utente

Il `create-access-key` comando seguente crea una chiave di accesso (ID della chiave di accesso e chiave di accesso segreta) per l'IAMutente denominatoBob.

```
aws iam create-access-key \  
  --user-name Bob
```

Output:

```
{  
  "AccessKey": {  
    "UserName": "Bob",  
    "Status": "Active",  
    "CreateDate": "2015-03-09T18:39:23.411Z",  
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYzEXAMPLEKEY",  
    "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"  
  }  
}
```

Conserva la chiave di accesso segreta in un luogo sicuro. Se viene persa, non può essere recuperata e dovrai creare una nuova chiave di accesso.

Per ulteriori informazioni, vedere [Gestione delle chiavi di accesso per IAM gli utenti](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [CreateAccessKey](#) in AWS CLI Command Reference.

## **create-account-alias**

Il seguente esempio di codice mostra come utilizzare `create-account-alias`.

### AWS CLI

Come Creare l'alias di un account

Il `create-account-alias` comando seguente crea l'alias `examplecorp` per il tuo AWS account.

```
aws iam create-account-alias \  
  --account-alias examplecorp
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta l'[ID AWS dell'account e il relativo alias nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, consulta [CreateAccountAlias AWS CLI](#) Command Reference.

## create-group

Il seguente esempio di codice mostra come utilizzare `create-group`.

### AWS CLI

Per creare un IAM gruppo

Il `create-group` comando seguente crea un IAM gruppo denominato `Admins`.

```
aws iam create-group \  
  --group-name Admins
```

Output:

```
{  
  "Group": {  
    "Path": "/",  
    "CreateDate": "2015-03-09T20:30:24.940Z",  
    "GroupId": "AIDGPMS9R04H3FEXAMPLE",  
    "Arn": "arn:aws:iam::123456789012:group/Admins",  
    "GroupName": "Admins"  
  }  
}
```

Per ulteriori informazioni, vedere [Creazione di gruppi di IAM utenti](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [CreateGroup](#) in AWS CLI Command Reference.

## create-instance-profile

Il seguente esempio di codice mostra come utilizzare `create-instance-profile`.

### AWS CLI

Come creare un profilo dell'istanza

Il comando `create-instance-profile` seguente crea un profilo dell'istanza denominato `Webserver`.

```
aws iam create-instance-profile \  
  --instance-profile-name Webserver
```

Output:

```
{  
  "InstanceProfile": {  
    "InstanceId": "AIPAJMBC7DLSPEXAMPLE",  
    "Roles": [],  
    "CreateDate": "2015-03-09T20:33:19.626Z",  
    "InstanceProfileName": "Webserver",  
    "Path": "/",  
    "Arn": "arn:aws:iam::123456789012:instance-profile/Webserver"  
  }  
}
```

Per aggiungere un ruolo a un profilo dell'istanza, usa il comando `add-role-to-instance-profile`.

Per ulteriori informazioni, consulta [Usare un IAM ruolo per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella Guida](#) per l'AWS IAM utente.

- Per API i dettagli, consulta AWS CLI Command [CreateInstanceProfile](#) Reference.

## create-login-profile

Il seguente esempio di codice mostra come utilizzare `create-login-profile`.

### AWS CLI

Per creare una password per un IAM utente

Per creare una password per un IAM utente, si consiglia di utilizzare il `--cli-input-json` parametro per passare un JSON file contenente la password. Utilizzando questo metodo, è possibile creare una password sicura con caratteri non alfanumerici. Può essere difficile creare una password con caratteri non alfanumerici quando la si passa come parametro della riga di comando.



Per utilizzare il `--cli-input-json` parametro, iniziate a utilizzare il `create-login-profile` comando con il `--generate-cli-skeleton` parametro, come nell'esempio seguente.

```
aws iam create-login-profile \  
  --generate-cli-skeleton > create-login-profile.json
```

Il comando precedente crea un JSON file chiamato `create-login-profile.json` che potete utilizzare per inserire le informazioni per un comando successivo `create-login-profile`. Per esempio:

```
{  
  "UserName": "Bob",  
  "Password": "&1-3a6u:RA0djs",  
  "PasswordResetRequired": true  
}
```

Quindi, per creare una password per un IAM utente, utilizzate nuovamente il `create-login-profile` comando, questa volta passando il `--cli-input-json` parametro per specificare il JSON file. Il `create-login-profile` comando seguente utilizza il `--cli-input-json` parametro con un JSON file chiamato `create-login-profile.json`.

```
aws iam create-login-profile \  
  --cli-input-json file://create-login-profile.json
```

Output:

```
{  
  "LoginProfile": {  
    "UserName": "Bob",  
    "CreateDate": "2015-03-10T20:55:40.274Z",  
    "PasswordResetRequired": true  
  }  
}
```

Se la nuova password viola la politica relativa alle password dell'account, il comando restituisce un errore. `PasswordPolicyViolation`

Per modificare la password di un utente che ne ha già una, usa `update-login-profile`. Per impostare una politica di password per l'account, usa il `update-account-password-policy` comando.

Se la politica sulla password dell'account lo consente, IAM gli utenti possono modificare le proprie password utilizzando il `change-password` comando.

Per ulteriori informazioni, vedere [Gestione delle password per IAM gli utenti nella Guida per l'AWS IAMutente](#).

- Per API i dettagli, vedere [CreateLoginProfile](#) in AWS CLI Command Reference.

## **create-open-id-connect-provider**

Il seguente esempio di codice mostra come utilizzare `create-open-id-connect-provider`.

### AWS CLI

Per creare un provider OpenID Connect (OIDC)

Per creare un provider OpenID Connect (OIDC), consigliamo di utilizzare il `--cli-input-json` parametro per passare un JSON file che contenga i parametri richiesti. Quando crei un OIDC provider, devi passarne il nome e URL devi iniziare con `https://`. URL Può essere difficile passarlo URL come parametro della riga di comando, perché i caratteri due punti (:) e barra (/) hanno un significato speciale in alcuni ambienti della riga di comando. L'utilizzo del `--cli-input-json` parametro consente di aggirare questa limitazione.

Per utilizzare il `--cli-input-json` parametro, iniziate a utilizzare il `create-open-id-connect-provider` comando con il `--generate-cli-skeleton` parametro, come nell'esempio seguente.

```
aws iam create-open-id-connect-provider \
  --generate-cli-skeleton > create-open-id-connect-provider.json
```

Il comando precedente crea un JSON file chiamato `create-open-id-connect-provider.json` che potete utilizzare per inserire le informazioni per un comando successivo. `create-open-id-connect-provider` Per esempio:

```
{
  "Url": "https://server.example.com",
  "ClientIDList": [
    "example-application-ID"
  ],
  "ThumbprintList": [
    "c3768084dfb3d2b68b7897bf5f565da8eEXAMPLE"
  ]
}
```

```
]
}
```

Quindi, per creare il provider OpenID Connect (OIDC), utilizzate nuovamente il `create-open-id-connect-provider` comando, questa volta passando il `--cli-input-json` parametro per specificare il JSON file. Il `create-open-id-connect-provider` comando seguente utilizza il `--cli-input-json` parametro con un JSON file chiamato `create-open-id-connect-provider.json`.

```
aws iam create-open-id-connect-provider \
  --cli-input-json file://create-open-id-connect-provider.json
```

Output:

```
{
  "OpenIDConnectProviderArn": "arn:aws:iam::123456789012:oidc-provider/
server.example.com"
}
```

Per ulteriori informazioni sui OIDC provider, vedete [Creating OpenID Connect \(OIDC\) identity provider nella Guida](#) per l'AWS IAMutente.

Per ulteriori informazioni su come ottenere impronte digitali per un OIDC provider, vedere [Ottenerne l'impronta personale per un provider di identità OpenID Connect](#) nella Guida per l'utente.AWS IAM

- Per i API dettagli, vedere in Command Reference. [CreateOpenIdConnectProvider](#)AWS CLI

## create-policy-version

Il seguente esempio di codice mostra come utilizzare `create-policy-version`.

AWS CLI

Per creare una nuova versione di una policy gestita

Questo esempio crea una nuova v2 versione della IAM policy che ARN è `arn:aws:iam::123456789012:policy/MyPolicy` e la rende la versione predefinita.

```
aws iam create-policy-version \
```

```
--policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
--policy-document file://NewPolicyVersion.json \  
--set-as-default
```

Output:

```
{  
  "PolicyVersion": {  
    "CreateDate": "2015-06-16T18:56:03.721Z",  
    "VersionId": "v2",  
    "IsDefaultVersion": true  
  }  
}
```

Per ulteriori informazioni, vedere [IAM Criteri di controllo delle versioni](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [CreatePolicyVersion](#) in AWS CLI Command Reference.

## create-policy

Il seguente esempio di codice mostra come utilizzare `create-policy`.

### AWS CLI

Esempio 1: Come creare una policy gestita dal cliente

Il comando seguente crea una policy gestita dal cliente denominata `my-policy`.

```
aws iam create-policy \  
  --policy-name my-policy \  
  --policy-document file://policy
```

Il file `policy` è un JSON documento nella cartella corrente che consente l'accesso in sola lettura alla `shared` cartella in un bucket Amazon S3 denominato `my-bucket`

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  

```

```

        "s3:Get*",
        "s3:List*"
    ],
    "Resource": [
        "arn:aws:s3:::my-bucket/shared/*"
    ]
}
]
}

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni sull'utilizzo dei file come input per i parametri delle stringhe, consulta [Specificare i valori dei parametri per la AWS CLI nella Guida per l'utente.AWS CLI](#)

Esempio 2: Come creare una policy gestita dal cliente con una descrizione

Il comando seguente crea una policy gestita dal cliente denominata `my-policy` con una descrizione non modificabile.

```

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"

```

Il file `policy.json` è un JSON documento nella cartella corrente che consente l'accesso a tutte le azioni Put, List e Get per un bucket Amazon S3 denominato `my-bucket`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket*",
        "s3:PutBucket*",
        "s3:GetBucket*"
      ],
      "Resource": [
        "arn:aws:s3:::my-bucket"
      ]
    }
  ]
}
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni sulle politiche basate sull'identità, consulta [Politiche basate sull'identità](#) e politiche basate sulle risorse nella Guida per l'utente.AWS IAM

Esempio 3: Come creare una policy gestita dal cliente con tag

Il comando seguente crea una policy gestita dal cliente denominata my-policy con tag. Questo esempio utilizza il parametro flag con i seguenti tag in formato: `--tags JSON '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value":`

"Seattle"}' In alternativa, il flag `--tags` può essere utilizzato con tag in formato abbreviato: `'Key=Department,Value=Accounting Key=Location,Value=Seattle'`.

```
aws iam create-policy \  
  --policy-name my-policy \  
  --policy-document file://policy.json \  
  --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",  
  "Value": "Seattle"}
```

Il file `policy.json` è un JSON documento nella cartella corrente che consente l'accesso a tutte le azioni Put, List e Get per un bucket Amazon S3 denominato. `my-bucket`

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:ListBucket*",  
        "s3:PutBucket*",  
        "s3:GetBucket*"  
      ],  
      "Resource": [  
        "arn:aws:s3:::my-bucket"  
      ]  
    }  
  ]  
}
```

Output:

```
{  
  "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"
    }
  ]
}
```

Per ulteriori informazioni sulle politiche di tagging, consulta [Tagging customer managed policy](#) nella Guida per l'utente AWS IAM

- Per API i dettagli, consulta AWS CLI Command [CreatePolicyReference](#).

## create-role

Il seguente esempio di codice mostra come utilizzare `create-role`.

### AWS CLI

Esempio 1: creare un IAM ruolo

Il comando `create-role` seguente crea un ruolo denominato `Test-Role` e collega una policy di attendibilità a tale ruolo.

```
aws iam create-role \
  --role-name Test-Role \
  --assume-role-policy-document file://Test-Role-Trust-Policy.json
```

Output:

```
{
  "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"
  }
}
```

La politica di fiducia è definita come JSON documento nel file `Test-Role-trust-policy.json`. (Il nome e l'estensione del file non hanno importanza.) La policy di attendibilità deve specificare un principale.

Per collegare una policy di autorizzazioni a un ruolo, usa il comando `put-role-policy`.

[Per ulteriori informazioni, consulta Creazione di ruoli nella Guida per l'utente. IAM AWS IAM](#)

Esempio 2: creare un IAM ruolo con una durata massima della sessione specificata

Il comando `create-role` seguente crea un ruolo denominato `Test-Role` e imposta una durata massima della sessione di 7200 secondi (2 ore).

```
aws iam create-role \
  --role-name Test-Role \
  --assume-role-policy-document file://Test-Role-Trust-Policy.json \
  --max-session-duration 7200
```

Output:

```
{
  "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"
        }
      ]
    }
  }
}
```

```

    ]
  }
}

```

Per ulteriori informazioni, vedere [Modifica della durata massima della sessione di un ruolo \(AWS API\)](#) nella Guida per l'AWS IAMutente.

### Esempio 3: creare un IAM ruolo con tag

Il comando seguente crea un IAM ruolo `Test-Role` con tag. Questo esempio utilizza il `--tags` parametro flag con i seguenti tag JSON in formato: `'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'`  
 In alternativa, il flag `--tags` può essere utilizzato con tag in formato abbreviato: `'Key=Department,Value=Accounting Key=Location,Value=Seattle'`.

```

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"}'

```

### Output:

```

{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Tagging IAM roles](#) nella User Guide.AWS IAM

- Per API i dettagli, vedere [CreateRole](#) in AWS CLI Command Reference.

## create-saml-provider

Il seguente esempio di codice mostra come utilizzare `create-saml-provider`.

### AWS CLI

Per creare un SAML provider

Questo esempio crea un nuovo SAML provider in IAM named `MySAMLProvider`. È descritto dal documento di SAML metadati che si trova nel file `SAMLMetaData.xml`.

```
aws iam create-saml-provider \
  --saml-metadata-document file://SAMLMetaData.xml \
  --name MySAMLProvider
```

Output:

```
{
  "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/MySAMLProvider"
}
```

Per ulteriori informazioni, consulta [Creazione di provider di IAM SAML identità](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [C reateSAMLProvider](#) in AWS CLI Command Reference.

## create-service-linked-role

Il seguente esempio di codice mostra come usare `create-service-linked-role`.

### AWS CLI

Come creare un ruolo collegato a un servizio

L'esempio seguente crea un ruolo collegato al servizio per il servizio AWS specificato e allega la descrizione specificata.

```
aws iam create-service-linked-role \  
  --aws-service-name lex.amazonaws.com \  
  --description "My service-linked role to support Lex"
```

Output:

```
{  
  "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"  
            ]  
          }  
        }  
      ]  
    }  
  }  
}
```

```
    }  
  }  
}
```

Per ulteriori informazioni, vedere [Utilizzo dei ruoli collegati ai servizi nella Guida](#) per l'utente AWS IAM

- Per API i dettagli, vedere [CreateServiceLinkedRole](#) in AWS CLI Command Reference.

## create-service-specific-credential

Il seguente esempio di codice mostra come utilizzare `create-service-specific-credential`.

### AWS CLI

Crea un set di credenziali specifiche del servizio per un utente

L'esempio seguente crea un nome utente e una password che possono essere utilizzati per accedere solo al servizio configurato.

```
aws iam create-service-specific-credential \  
  --user-name sofia \  
  --service-name codecommit.amazonaws.com
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Create Git credenziali per HTTPS le connessioni CodeCommit](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [CreateServiceSpecificCredential](#) in AWS CLI Command Reference.

## create-user

Il seguente esempio di codice mostra come utilizzare `create-user`.

### AWS CLI

Esempio 1: creare un IAM utente

Il `create-user` comando seguente crea un IAM utente denominato Bob nell'account corrente.

```
aws iam create-user \  
  --user-name Bob
```

Output:

```
{  
  "User": {  
    "UserName": "Bob",  
    "Path": "/",  
    "CreateDate": "2023-06-08T03:20:41.270Z",  
    "UserId": "AIDAIOSFODNN7EXAMPLE",  
    "Arn": "arn:aws:iam::123456789012:user/Bob"  
  }  
}
```

Per ulteriori informazioni, consulta [Creazione di un IAM utente nel tuo AWS account](#) nella Guida AWS IAM per l'utente.

Esempio 2: creare un IAM utente in un percorso specificato

Il `create-user` comando seguente crea un IAM utente denominato Bob nel percorso specificato.

```
aws iam create-user \  
  --user-name Bob \  
  --path /division_abc/subdivision_xyz/
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consultate [IAM gli identificatori](#) nella Guida per l'AWS IAM utente.

### Esempio 3: creare un IAM utente con tag

Il `create-user` comando seguente crea un IAM utente denominato Bob con tag. Questo esempio utilizza il `--tags` parametro flag con i seguenti tag JSON in formato: `'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'` In alternativa, il flag `--tags` può essere utilizzato con tag in formato abbreviato: `'Key=Department,Value=Accounting Key=Location,Value=Seattle'`.

```

aws iam create-user \
  --user-name Bob \
  --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
"Value": "Seattle"}'

```

Output:

```

{
  "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"
      }
    ]
  }
}

```

```
}  
}
```

Per ulteriori informazioni, consulta [Taggare IAM gli utenti nella Guida](#) per l'AWS IAMutente.

Esempio 3: creare un IAM utente con un limite di autorizzazioni impostato

Il `create-user` comando seguente crea un IAM utente denominato Bob con il limite di autorizzazioni di AmazonS3. FullAccess

```
aws iam create-user \  
  --user-name Bob \  
  --permissions-boundary arn:aws:iam::aws:policy/AmazonS3FullAccess
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Limiti delle autorizzazioni per le entità nella Guida per l'IAM](#)utente.AWS IAM

- Per API i dettagli, vedere [CreateUser](#)in AWS CLI Command Reference.

## **create-virtual-mfa-device**

Il seguente esempio di codice mostra come utilizzare `create-virtual-mfa-device`.

AWS CLI

Per creare un MFA dispositivo virtuale



Questo esempio crea un nuovo MFA dispositivo virtuale chiamato `BobsMFADevice`. Crea un file che contiene le informazioni di bootstrap richiamate `QRCode.png` e le inserisce nella `C:/` directory. Il metodo bootstrap utilizzato in questo esempio è `QRCodePNG`

```
aws iam create-virtual-mfa-device \  
  --virtual-mfa-device-name BobsMFADevice \  
  --outfile C:/QRCode.png \  
  --bootstrap-method QRCodePNG
```

Output:

```
{  
  "VirtualMFADevice": {  
    "SerialNumber": "arn:aws:iam::210987654321:mfa/BobsMFADevice"  
  }  
}
```

Per ulteriori informazioni, vedete [Using Multi-Factor Authentication \(MFA\) AWS nella Guida per l'AWS IAMutente](#).

- Per API i dettagli, vedere [CreateVirtualMfaDevice](#) in AWS CLI Command Reference.

## **deactivate-mfa-device**

Il seguente esempio di codice mostra come utilizzare `deactivate-mfa-device`.

AWS CLI

Per disattivare un dispositivo MFA

Questo comando disattiva il MFA dispositivo virtuale associato ARN `arn:aws:iam::210987654321:mfa/BobsMFADevice` all'utente. Bob

```
aws iam deactivate-mfa-device \  
  --user-name Bob \  
  --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Using Multi-Factor Authentication \(MFA\) AWS nella Guida per l'AWS IAMutente](#).

- Per API i dettagli, vedere [DeactivateMfaDevice](#) in AWS CLI Command Reference.

## decode-authorization-message

Il seguente esempio di codice mostra come utilizzare `decode-authorization-message`.

### AWS CLI

Per decodificare un messaggio di errore di autorizzazione

L'esempio seguente decodifica il messaggio restituito dalla EC2 console quando si tenta di avviare un'istanza senza le autorizzazioni richieste.

```
aws sts decode-authorization-message \
  --encoded-message LxzA8VEjEvu-s0TTt3PgYCXik9Yak0qsrFJGRZR98xNcyWAxwRq14xIvd-
  npzbgTevuufCTbjeBAaDARg9cbTK1rJbg3awM33o-Vy3ebPErE2-
  mWR9hVYdvX-0zKgV0WF9pWjZaJSMqxB-aLXo-I_8TTvBq88x8IFPbMArNdpu0IjxDjzf22PF3S0E3XvIQ-
  _PE00aUqHCCcsSrFtvxm6yQD1nbm6VTIVrfa0Bzy8LsoMo7SjIaJ2r5vph6SY5vCCwg6o2JKe3hIHTa8zRrDbZSFMkcX
  Xx9AYAAIr6bhcis7C__bZh4dLAAWooHFGKgfoJcWGWgdzgbu9hWyVvKTpeot5hsb8qANYjJRCPXTKpi6PZfdijIkwb6g
```

L'output è formattato come una stringa di testo a riga singola che può essere analizzata con qualsiasi elaboratore di JSON testo. 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:epsOptimized\", \"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-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/*\"}]}]}]"}"
```

```
}
```

Per ulteriori informazioni, vedi [Come posso decodificare un messaggio di errore di autorizzazione dopo aver ricevuto un errore "UnauthorizedOperation" durante l'avvio di un'istanza? EC2 in AWS](#) Re:post.

- Per API i dettagli, vedere [DecodeAuthorizationMessage](#) in AWS CLI Command Reference.

## **delete-access-key**

Il seguente esempio di codice mostra come utilizzare `delete-access-key`.

### AWS CLI

Per eliminare una chiave di accesso per un IAM utente

Il `delete-access-key` comando seguente elimina la chiave di accesso specificata (ID della chiave di accesso e chiave di accesso segreta) per l'IAM utente denominato Bob.

```
aws iam delete-access-key \  
  --access-key-id AKIDPMS9R04H3FEXAMPLE \  
  --user-name Bob
```

Questo comando non produce alcun output.

Per elencare le chiavi di accesso definite per un IAM utente, utilizzare il `list-access-keys` comando.

Per ulteriori informazioni, vedere [Gestione delle chiavi di accesso per IAM gli utenti](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [DeleteAccessKey](#) in AWS CLI Command Reference.

## **delete-account-alias**

Il seguente esempio di codice mostra come utilizzare `delete-account-alias`.

### AWS CLI

Come eliminare l'alias di un account

Il comando `delete-account-alias` seguente rimuove l'alias `mycompany` per l'account corrente.

```
aws iam delete-account-alias \  
--account-alias mycompany
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta l'[ID AWS dell'account e il relativo alias](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, consulta [DeleteAccountAlias AWS CLI Command Reference](#).

## delete-account-password-policy

Il seguente esempio di codice mostra come utilizzare `delete-account-password-policy`.

### AWS CLI

Per eliminare la politica corrente in materia di password dell'account

Il `delete-account-password-policy` comando seguente rimuove la politica relativa alle password per l'account corrente.

```
aws iam delete-account-password-policy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Impostazione di una politica di password dell'account per IAM gli utenti](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [DeleteAccountPasswordPolicy](#) in AWS CLI Command Reference.

## delete-group-policy

Il seguente esempio di codice mostra come utilizzare `delete-group-policy`.

### AWS CLI

Per eliminare una politica da un IAM gruppo

Il comando `delete-group-policy` seguente elimina la policy denominata `ExamplePolicy` dal gruppo denominato `Admins`.

```
aws iam delete-group-policy \  
  --group-name Admins \  
  --policy-name ExamplePolicy
```

Questo comando non produce alcun output.

Per visualizzare le policy collegate a un gruppo, usa il comando `list-group-policies`.

Per ulteriori informazioni, vedere [Gestione delle IAM politiche](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [DeleteGroupPolicy](#) in AWS CLI Command Reference.

## **delete-group**

Il seguente esempio di codice mostra come utilizzare `delete-group`.

### AWS CLI

Per eliminare un IAM gruppo

Il `delete-group` comando seguente elimina un IAM gruppo denominato `MyTestGroup`.

```
aws iam delete-group \  
  --group-name MyTestGroup
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Eliminazione di un gruppo di IAM utenti nella Guida](#) per l'AWS IAM utente.

- Per API i dettagli, vedere [DeleteGroup](#) in AWS CLI Command Reference.

## **delete-instance-profile**

Il seguente esempio di codice mostra come utilizzare `delete-instance-profile`.

### AWS CLI

Come eliminare un profilo dell'istanza

Il comando `delete-instance-profile` seguente elimina un profilo dell'istanza denominato `ExampleInstanceProfile`.

```
aws iam delete-instance-profile \  
  --instance-profile-name ExampleInstanceProfile
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Using instance profiles \(Utilizzo dei profili di istanza\)](#) nella Guida per AWS IAM l'utente.

- Per API i dettagli, vedere [DeleteInstanceProfile](#) in AWS CLI Command Reference.

## delete-login-profile

Il seguente esempio di codice mostra come utilizzare `delete-login-profile`.

### AWS CLI

Per eliminare una password per un IAM utente

Il `delete-login-profile` comando seguente elimina la password per l'IAM utente denominato `Bob`.

```
aws iam delete-login-profile \  
  --user-name Bob
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Gestione delle password per IAM gli utenti nella Guida per l'AWS IAM utente](#).

- Per API i dettagli, vedere [DeleteLoginProfile](#) in AWS CLI Command Reference.

## delete-open-id-connect-provider

Il seguente esempio di codice mostra come utilizzare `delete-open-id-connect-provider`.

### AWS CLI

Per eliminare un provider di IAM identità OpenID Connect

Questo esempio elimina il IAM OIDC provider che si connette al provider.  
`example.oidcprovider.com`

```
aws iam delete-open-id-connect-provider \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
example.oidcprovider.com
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creazione di provider di identità OpenID Connect \(OIDC\) nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, consultate [DeleteOpenIdConnectProvider AWS CLI](#) Command Reference.

## delete-policy-version

Il seguente esempio di codice mostra come utilizzare `delete-policy-version`.

### AWS CLI

Per eliminare una versione di una politica gestita

Questo esempio elimina la versione identificata come v2 dalla policy cui ARN è `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam delete-policy-version \  
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
  --version-id v2
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Politiche e autorizzazioni IAM nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [DeletePolicyVersion](#) in AWS CLI Command Reference.

## delete-policy

Il seguente esempio di codice mostra come utilizzare `delete-policy`.

### AWS CLI

Per eliminare una IAM politica

Questo esempio elimina la policy cui ARN è `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam delete-policy \  
  --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Politiche e autorizzazioni IAM nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [DeletePolicy](#) in AWS CLI Command Reference.

## **delete-role-permissions-boundary**

Il seguente esempio di codice mostra come utilizzare `delete-role-permissions-boundary`.

### AWS CLI

Per eliminare un limite di autorizzazioni da un ruolo IAM

L'`delete-role-permissions-boundary` esempio seguente elimina il limite delle autorizzazioni per il ruolo specificato. IAM Per applicare un limite di autorizzazioni a un ruolo, utilizzate il comando. `put-role-permissions-boundary`

```
aws iam delete-role-permissions-boundary \  
  --role-name lambda-application-role
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Politiche e autorizzazioni IAM nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [DeleteRolePermissionsBoundary](#) in AWS CLI Command Reference.

## **delete-role-policy**

Il seguente esempio di codice mostra come utilizzare `delete-role-policy`.

### AWS CLI

Per rimuovere una politica da un IAM ruolo

Il comando `delete-role-policy` seguente rimuove la policy denominata `ExamplePolicy` dal ruolo denominato `Test-Role`.



```
aws iam delete-role-policy \  
  --role-name Test-Role \  
  --policy-name ExamplePolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Modificare un ruolo](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [DeleteRolePolicy](#) in AWS CLI Command Reference.

## delete-role

Il seguente esempio di codice mostra come utilizzare `delete-role`.

### AWS CLI

Per eliminare un IAM ruolo

Il comando `delete-role` seguente rimuove il ruolo denominato `Test-Role`.

```
aws iam delete-role \  
  --role-name Test-Role
```

Questo comando non produce alcun output.

Per poter eliminare un ruolo, devi prima rimuovere il ruolo da qualunque profilo dell'istanza (`remove-role-from-instance-profile`), scollegare eventuali policy gestite (`detach-role-policy`) ed eliminare tutte le policy inline collegate al ruolo (`delete-role-policy`).

Per ulteriori informazioni, consulta [Creazione di IAM ruoli](#) e [Utilizzo dei profili di istanza](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [DeleteRole](#) in AWS CLI Command Reference.

## delete-saml-provider

Il seguente esempio di codice mostra come utilizzare `delete-saml-provider`.

### AWS CLI

Per eliminare un SAML provider

Questo esempio elimina il provider IAM SAML 2.0 il cui ARN è `arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider`.

```
aws iam delete-saml-provider \  
--saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Creazione di provider di IAM SAML identità](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [DeleteSAMLProvider](#) in AWS CLI Command Reference.

## **delete-server-certificate**

Il seguente esempio di codice mostra come utilizzare `delete-server-certificate`.

### AWS CLI

Per eliminare un certificato del server dal tuo AWS account

Il `delete-server-certificate` comando seguente rimuove il certificato del server specificato dal tuo AWS account.

```
aws iam delete-server-certificate \  
--server-certificate-name myUpdatedServerCertificate
```

Questo comando non produce alcun output.

Per elencare i certificati server disponibili nel tuo AWS account, usa il `list-server-certificates` comando.

Per ulteriori informazioni, consulta [la sezione Gestione dei certificati del server IAM nella Guida AWS IAM per l'utente](#).

- Per API i dettagli, vedere [DeleteServerCertificate](#) in AWS CLI Command Reference.

## **delete-service-linked-role**

Il seguente esempio di codice mostra come utilizzare `delete-service-linked-role`.

## AWS CLI

Come eliminare un ruolo collegato a un servizio

L'esempio `delete-service-linked-role` seguente elimina il ruolo collegato al servizio specificato che non è più necessario. L'eliminazione avviene in modo asincrono. Puoi anche controllare lo stato dell'eliminazione e confermare quando è stata completata utilizzando il comando `get-service-linked-role-deletion-status`.

```
aws iam delete-service-linked-role \  
  --role-name AWSServiceRoleForLexBots
```

Output:

```
{  
  "DeletionTaskId": "task/aws-service-role/lex.amazonaws.com/  
  AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE"  
}
```

Per ulteriori informazioni, vedere [Utilizzo dei ruoli collegati ai servizi nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [DeleteServiceLinkedRole](#) in AWS CLI Command Reference.

## `delete-service-specific-credential`

Il seguente esempio di codice mostra come utilizzare `delete-service-specific-credential`.

### AWS CLI

Esempio 1: eliminare una credenziale specifica del servizio per l'utente richiedente

L'`delete-service-specific-credential`esempio seguente elimina la credenziale specifica del servizio specificata per l'utente che effettua la richiesta. `service-specific-credential-id`Viene fornita al momento della creazione della credenziale ed è possibile recuperarla utilizzando il comando. `list-service-specific-credentials`

```
aws iam delete-service-specific-credential \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Questo comando non produce alcun output.

Esempio 2: Eliminare una credenziale specifica del servizio per un utente specificato

L'`delete-service-specific-credential` seguente elimina la credenziale specifica del servizio specificata per l'utente specificato. `service-specific-credential-id` Viene fornita al momento della creazione della credenziale ed è possibile recuperarla utilizzando il comando `list-service-specific-credentials`

```
aws iam delete-service-specific-credential \  
  --user-name sofia \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Create Git credenziali per HTTPS le connessioni CodeCommit](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [DeleteServiceSpecificCredential](#) in AWS CLI Command Reference.

## `delete-signing-certificate`

Il seguente esempio di codice mostra come utilizzare `delete-signing-certificate`.

### AWS CLI

Per eliminare un certificato di firma per un IAM utente

Il `delete-signing-certificate` comando seguente elimina il certificato di firma specificato per l'IAM utente denominato Bob.

```
aws iam delete-signing-certificate \  
  --user-name Bob \  
  --certificate-id TA7SMP42TDN5Z260BPJE7EXAMPLE
```

Questo comando non produce alcun output.

Per ottenere l'ID per un certificato di firma, usa il `list-signing-certificates` comando.

Per ulteriori informazioni, consulta [Gestire i certificati di firma](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [DeleteSigningCertificate AWS CLI Command Reference](#).

## delete-ssh-public-key

Il seguente esempio di codice mostra come utilizzare `delete-ssh-public-key`.

### AWS CLI

Per eliminare una chiave SSH pubblica allegata a un IAM utente

Il `delete-ssh-public-key` comando seguente elimina la chiave SSH pubblica specificata allegata all'IAM utente `sofia`.

```
aws iam delete-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-id APKA123456789EXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Use SSH keys and SSH with CodeCommit](#) nella Guida per l'IAM utente.

- Per API i dettagli, vedere [DeleteSshPublicKey](#) in AWS CLI Command Reference.

## delete-user-permissions-boundary

Il seguente esempio di codice mostra come utilizzare `delete-user-permissions-boundary`.

### AWS CLI

Per eliminare un limite di autorizzazioni da un utente IAM

L'esempio seguente elimina il limite delle autorizzazioni associato all'utente denominato `intern`. Per applicare un limite di autorizzazioni a un utente, utilizzate il comando `put-user-permissions-boundary`.

```
aws iam delete-user-permissions-boundary \  
  --user-name intern
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Politiche e autorizzazioni IAM nella Guida](#) per l'IAM utente.

- Per API i dettagli, vedere [DeleteUserPermissionsBoundary](#) in AWS CLI Command Reference.

## delete-user-policy

Il seguente esempio di codice mostra come utilizzare `delete-user-policy`.

### AWS CLI

Per rimuovere una politica da un IAM utente

Il `delete-user-policy` comando seguente rimuove la politica specificata dal nome IAM utente `Bob`.

```
aws iam delete-user-policy \  
  --user-name Bob \  
  --policy-name ExamplePolicy
```

Questo comando non produce alcun output.

Per ottenere un elenco di politiche per un IAM utente, utilizzare il `list-user-policies` comando.

Per ulteriori informazioni, consulta [Creazione di un IAM utente nel tuo AWS account](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, consulta [DeleteUserPolicy AWS CLI Command Reference](#).

## delete-user

Il seguente esempio di codice mostra come utilizzare `delete-user`.

### AWS CLI

Per eliminare un IAM utente

Il `delete-user` comando seguente rimuove l'IAM utente denominato `Bob` dall'account corrente.

```
aws iam delete-user \  
  --user-name Bob
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Eliminazione di un IAM utente](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [DeleteUser](#) in AWS CLI Command Reference.

## delete-virtual-mfa-device

Il seguente esempio di codice mostra come utilizzare `delete-virtual-mfa-device`.

### AWS CLI

Per rimuovere un MFA dispositivo virtuale

Il `delete-virtual-mfa-device` comando seguente rimuove il MFA dispositivo specificato dall'account corrente.

```
aws iam delete-virtual-mfa-device \  
  --serial-number arn:aws:iam::123456789012:mfa/MFATest
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [MFADisattivazione dei dispositivi nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [DeleteVirtualMfaDevice](#) in AWS CLI Command Reference.

## detach-group-policy

Il seguente esempio di codice mostra come utilizzare `detach-group-policy`.

### AWS CLI

Per scollegare una politica da un gruppo

Questo esempio rimuove la politica gestita con il ARN `arn:aws:iam::123456789012:policy/TesterAccessPolicy` dal gruppo chiamato `Testers`.

```
aws iam detach-group-policy \  
  --group-name Testers \  
  --policy-arn arn:aws:iam::123456789012:policy/TesterAccessPolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Gestione dei gruppi di IAM utenti](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [DetachGroupPolicy](#) in AWS CLI Command Reference.

## detach-role-policy

Il seguente esempio di codice mostra come utilizzare `detach-role-policy`.

### AWS CLI

Come scollegare una policy da un ruolo

Questo esempio rimuove la politica gestita con il ARN `arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy` dal ruolo chiamato `FedTesterRole`.

```
aws iam detach-role-policy \  
  --role-name FedTesterRole \  
  --policy-arn arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Modifica di un ruolo](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [DetachRolePolicy](#) in AWS CLI Command Reference.

## detach-user-policy

Il seguente esempio di codice mostra come utilizzare `detach-user-policy`.

### AWS CLI

Come scollegare una policy da un utente

Questo esempio rimuove la policy gestita con l'ARN `arn:aws:iam::123456789012:policy/TesterPolicy` utente `Bob`.

```
aws iam detach-user-policy \  
  --user-name Bob \  
  --policy-arn arn:aws:iam::123456789012:policy/TesterPolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Modifica delle autorizzazioni per un IAM utente](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [DetachUserPolicy](#) in AWS CLI Command Reference.



## enable-mfa-device

Il seguente esempio di codice mostra come utilizzare `enable-mfa-device`.

### AWS CLI

Per abilitare un MFA dispositivo

Dopo aver utilizzato il `create-virtual-mfa-device` comando per creare un nuovo MFA dispositivo virtuale, è possibile assegnare il MFA dispositivo a un utente. L'`enable-mfa-device` esempio seguente assegna il MFA dispositivo con il numero di serie `arn:aws:iam::210987654321:mfa/BobsMFADevice` all'utente. Bob Il comando sincronizza inoltre il dispositivo con il dispositivo virtuale AWS MFA includendo i primi due codici in sequenza.

```
aws iam enable-mfa-device \  
  --user-name Bob \  
  --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \  
  --authentication-code1 123456 \  
  --authentication-code2 789012
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Abilitazione di un dispositivo virtuale di autenticazione a più fattori \(MFA\) nella Guida](#) per l'AWS IAM utente.

- Per API i dettagli, vedere [EnableMfaDevice](#) in AWS CLI Command Reference.

## generate-credential-report

Il seguente esempio di codice mostra come utilizzare `generate-credential-report`.

### AWS CLI

Come generare un report delle credenziali

L'esempio seguente tenta di generare un rapporto sulle credenziali per l' AWS account.

```
aws iam generate-credential-report
```

Output:

```
{
```

```
"State": "STARTED",
  "Description": "No report exists. Starting a new report generation task"
}
```

Per ulteriori informazioni, consulta [Ottenerne i report sulle credenziali per il tuo AWS account](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, consulta [GenerateCredentialReport AWS CLI](#) Command Reference.

## generate-organizations-access-report

Il seguente esempio di codice mostra come utilizzare `generate-organizations-access-report`.

### AWS CLI

Esempio 1: generare un rapporto di accesso per una radice in un'organizzazione

L'`generate-organizations-access-report` esempio seguente avvia un processo in background per creare un rapporto di accesso per la radice specificata in un'organizzazione. È possibile visualizzare il report dopo averlo creato eseguendo il `get-organizations-access-report` comando.

```
aws iam generate-organizations-access-report \
  --entity-path o-4fxmpl1t198/r-c3xb
```

Output:

```
{
  "JobId": "a8b6c06f-aaa4-8xmp-28bc-81da71836359"
}
```

Esempio 2: generare un rapporto di accesso per un account in un'organizzazione

L'`generate-organizations-access-report` esempio seguente avvia un processo in background per creare un rapporto di accesso per l'ID dell'account 123456789012 nell'organizzazione `o-4fxmpl1t198`. È possibile visualizzare il report dopo averlo creato eseguendo il `get-organizations-access-report` comando.

```
aws iam generate-organizations-access-report \
```

```
--entity-path o-4fxmpl1t198/r-c3xb/123456789012
```

Output:

```
{
  "JobId": "14b6c071-75f6-2xmp-fb77-faf6fb4201d2"
}
```

Esempio 3: generare un rapporto di accesso per un account in un'unità organizzativa di un'organizzazione

L'generate-organizations-access-report esempio seguente avvia un processo in background per creare un rapporto di accesso per l'ID dell'account 234567890123 ou-c3xb-lmu7j2yg nell'unità organizzativa dell'organizzazione o-4fxmpl1t198. È possibile visualizzare il report dopo averlo creato eseguendo il get-organizations-access-report comando.

```
aws iam generate-organizations-access-report \
  --entity-path o-4fxmpl1t198/r-c3xb/ou-c3xb-lmu7j2yg/234567890123
```

Output:

```
{
  "JobId": "2eb6c2e6-0xmp-ec04-1425-c937916a64af"
}
```

Per ottenere dettagli sulle radici e sulle unità organizzative dell'organizzazione, utilizza i `organizations list-organizational-units-for-parent` comandi `organizations list-roots` and.

Per ulteriori informazioni, vedere [Perfezionamento delle autorizzazioni nell' AWS uso delle informazioni dell'ultimo accesso nella Guida](#) per l'AWS IAM utente.

- Per API i dettagli, vedere [GenerateOrganizationsAccessReport](#) in AWS CLI Command Reference.

## generate-service-last-accessed-details

Il seguente esempio di codice mostra come utilizzare `generate-service-last-accessed-details`.

## AWS CLI

Esempio 1: generare un rapporto di accesso al servizio per una politica personalizzata

L'`generate-service-last-accessed-details` seguente avvia un processo in background per generare un report che elenca i servizi a cui accedono IAM gli utenti e altre entità con una politica personalizzata denominata `intern-boundary`. È possibile visualizzare il report dopo averlo creato eseguendo il `get-service-last-accessed-details` comando.

```
aws iam generate-service-last-accessed-details \  
--arn arn:aws:iam::123456789012:policy/intern-boundary
```

Output:

```
{  
  "JobId": "2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc"  
}
```

Esempio 2: generare un rapporto di accesso al servizio per la `AdministratorAccess` politica AWS gestita

L'`generate-service-last-accessed-details` seguente avvia un processo in background per generare un report che elenca i servizi a cui accedono IAM gli utenti e altre entità con la `AdministratorAccess` politica AWS gestita. È possibile visualizzare il report dopo averlo creato eseguendo il `get-service-last-accessed-details` comando.

```
aws iam generate-service-last-accessed-details \  
--arn arn:aws:iam::aws:policy/AdministratorAccess
```

Output:

```
{  
  "JobId": "78b6c2ba-d09e-6xmp-7039-ecde30b26916"  
}
```

Per ulteriori informazioni, vedere [Perfezionamento delle autorizzazioni nell' AWS utilizzo delle informazioni dell'ultimo accesso nella Guida](#) per l'AWS IAM utente.

- Per API i dettagli, vedere [GenerateServiceLastAccessedDetails](#) in AWS CLI Command Reference.

## get-access-key-last-used

Il seguente esempio di codice mostra come utilizzare `get-access-key-last-used`.

### AWS CLI

Come recuperare informazioni sull'ultimo utilizzo della chiave di accesso specificata

L'esempio seguente recupera informazioni sull'ultimo utilizzo della chiave di accesso ABCDEXAMPLE.

```
aws iam get-access-key-last-used \
  --access-key-id ABCDEXAMPLE
```

Output:

```
{
  "UserName": "Bob",
  "AccessKeyLastUsed": {
    "Region": "us-east-1",
    "ServiceName": "iam",
    "LastUsedDate": "2015-06-16T22:45:00Z"
  }
}
```

Per ulteriori informazioni, vedere [Gestione delle chiavi di accesso per IAM gli utenti](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [GetAccessKeyLastUsed](#) in AWS CLI Command Reference.

## get-account-authorization-details

Il seguente esempio di codice mostra come utilizzare `get-account-authorization-details`.

### AWS CLI

Per elencare un AWS account IAM utenti, gruppi, ruoli e politiche

Il `get-account-authorization-details` comando seguente restituisce informazioni su tutti IAM gli utenti, i gruppi, i ruoli e le politiche dell' AWS account.

```
aws iam get-account-authorization-details
```

## Output:

```
{
  "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": "Stmnt1381777017000",
                        "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": "EXAMPLEkakov9BCuUNFDtxWSyetzYwEx2ADc8dnzfvERF5S6YMvXKx41t6gCl/
    eeaCX3Jo94/bKqezEAg8TEVS99EKFLxm3jtbpl25FDWEXAMPLE",
    "IsTruncated": true
}

```

Per ulteriori informazioni, consulta le [linee guida per il controllo di AWS sicurezza](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [GetAccountAuthorizationDetails](#) in AWS CLI Command Reference.

## get-account-password-policy

Il seguente esempio di codice mostra come utilizzare `get-account-password-policy`.

### AWS CLI

Come visualizzare la policy delle password dell'account corrente

Il comando `get-account-password-policy` seguente visualizza dettagli sulla policy delle password per l'account corrente.

```
aws iam get-account-password-policy
```

Output:

```
{
  "PasswordPolicy": {
    "AllowUsersToChangePassword": false,
    "RequireLowercaseCharacters": false,
    "RequireUppercaseCharacters": false,
    "MinimumPasswordLength": 8,
    "RequireNumbers": true,
    "RequireSymbols": true
  }
}
```

Se non è definita alcuna policy delle password per l'account, il comando restituisce un errore `NoSuchEntity`.

Per ulteriori informazioni, vedere [Impostazione di una politica di password dell'account per IAM gli utenti](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [GetAccountPasswordPolicy](#) in AWS CLI Command Reference.

## **get-account-summary**

Il seguente esempio di codice mostra come utilizzare `get-account-summary`.

AWS CLI

Per ottenere informazioni sull'utilizzo delle IAM entità e sulle IAM quote nell'account corrente

Il `get-account-summary` comando seguente restituisce informazioni sull'utilizzo corrente dell'IAM entità e sulle quote correnti delle IAM entità nell'account.

```
aws iam get-account-summary
```

Output:

```
{
```

```
"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
}
```

Per ulteriori informazioni sulle limitazioni delle entità, consulta [IAMe sulle AWS STS quote nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [GetAccountSummary](#) in AWS CLI Command Reference.

## get-context-keys-for-custom-policy

Il seguente esempio di codice mostra come utilizzare `get-context-keys-for-custom-policy`.

### AWS CLI

Esempio 1: per elencare le chiavi di contesto a cui fanno riferimento una o più JSON politiche personalizzate fornite come parametro nella riga di comando

Il `get-context-keys-for-custom-policy` comando seguente analizza ogni politica fornita ed elenca le chiavi di contesto utilizzate da tali politiche. Utilizzate questo comando per

identificare quali valori chiave di contesto è necessario fornire per utilizzare correttamente i comandi `simulate-custom-policy` del simulatore di politiche e `simulate-custom-policy`. È inoltre possibile recuperare l'elenco delle chiavi di contesto utilizzate da tutte le politiche associate da un IAM utente o da un ruolo utilizzando il `get-context-keys-for-custom-policy` comando. I valori dei parametri che iniziano con `file://` indicano al comando di leggere il file e di utilizzare il contenuto come valore per il parametro anziché il nome del file stesso.

```
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"}}}}'
```

Output:

```
{
  "ContextKeyNames": [
    "aws:username",
    "aws:CurrentTime"
  ]
}
```

Esempio 2: elencare le chiavi di contesto a cui fanno riferimento una o più JSON politiche personalizzate fornite come input del file

Il `get-context-keys-for-custom-policy` comando seguente è lo stesso dell'esempio precedente, tranne per il fatto che le politiche vengono fornite in un file anziché come parametro. Poiché il comando prevede un JSON elenco di stringhe e non un elenco di JSON strutture, il file deve essere strutturato come segue, sebbene sia possibile comprimerlo in un unico file.

```
[
  "Policy1",
  "Policy2"
]
```

Ad esempio, un file che contiene la politica dell'esempio precedente deve avere l'aspetto seguente. È necessario evitare ogni virgoletta doppia incorporata nella stringa di policy facendola precedere da una barra rovesciata».

```
[ {"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"}}} ] }
```

Questo file può quindi essere inviato al seguente comando.

```
aws iam get-context-keys-for-custom-policy \
  --policy-input-list file://policyfile.json
```

Output:

```
{
  "ContextKeyNames": [
    "aws:username",
    "aws:CurrentTime"
  ]
}
```

Per ulteriori informazioni, vedere [Using the IAM Policy Simulator \(AWS CLI and AWS API\)](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [GetContextKeysForCustomPolicy](#) in AWS CLI Command Reference.

## get-context-keys-for-principal-policy

Il seguente esempio di codice mostra come utilizzare `get-context-keys-for-principal-policy`.

### AWS CLI

Per elencare le chiavi di contesto a cui fanno riferimento tutte le politiche associate a un principale IAM

Il `get-context-keys-for-principal-policy` comando seguente recupera tutte le politiche allegare all'utente `saarvi` e ai gruppi di cui è membro. Quindi analizza ciascuna di esse ed elenca le chiavi di contesto utilizzate da tali politiche. Utilizzate questo comando per identificare i valori delle chiavi di contesto che dovete fornire per utilizzare correttamente `simulate-principal-policy` i comandi `simulate-custom-policy` and. È inoltre possibile recuperare



l'elenco delle chiavi di contesto utilizzate da una JSON politica arbitraria utilizzando il `get-context-keys-for-custom-policy` comando.

```
aws iam get-context-keys-for-principal-policy \  
  --policy-source-arn arn:aws:iam::123456789012:user/saanvi
```

Output:

```
{  
  "ContextKeyNames": [  
    "aws:username",  
    "aws:CurrentTime"  
  ]  
}
```

Per ulteriori informazioni, vedere [Using the IAM Policy Simulator \(AWS CLI and AWS API\)](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [GetContextKeysForPrincipalPolicy](#) in AWS CLI Command Reference.

## get-credential-report

Il seguente esempio di codice mostra come utilizzare `get-credential-report`.

### AWS CLI

Come ottenere un report delle credenziali

Questo esempio apre il report restituito e lo invia alla pipeline come array di righe di testo.

```
aws iam get-credential-report
```

Output:

```
{  
  "GeneratedTime": "2015-06-17T19:11:50Z",  
  "ReportFormat": "text/csv"  
}
```

Per ulteriori informazioni, consulta [Ottenere i report sulle credenziali per il tuo AWS account](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, consulta [GetCredentialReport AWS CLI Command Reference](#).

## get-group-policy

Il seguente esempio di codice mostra come utilizzare `get-group-policy`.

### AWS CLI

Per ottenere informazioni su una politica associata a un IAM gruppo

Il `get-group-policy` comando seguente ottiene informazioni sulla politica specificata allegata al gruppo denominato `Test-Group`.

```
aws iam get-group-policy \  
  --group-name Test-Group \  
  --policy-name S3-ReadOnly-Policy
```

Output:

```
{  
  "GroupName": "Test-Group",  
  "PolicyDocument": {  
    "Statement": [  
      {  
        "Action": [  
          "s3:Get*",  
          "s3:List*"  
        ],  
        "Resource": "*",  
        "Effect": "Allow"  
      }  
    ]  
  },  
  "PolicyName": "S3-ReadOnly-Policy"  
}
```

Per ulteriori informazioni, vedere [Gestione delle IAM politiche](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [GetGroupPolicy](#) in AWS CLI Command Reference.

## get-group

Il seguente esempio di codice mostra come utilizzare `get-group`.

### AWS CLI

Per creare un IAM gruppo

Questo esempio restituisce dettagli sul IAM gruppo `Admins`.

```
aws iam get-group \
  --group-name Admins
```

Output:

```
{
  "Group": {
    "Path": "/",
    "CreateDate": "2015-06-16T19:41:48Z",
    "GroupId": "AIDGPMS9R04H3FEXAMPLE",
    "Arn": "arn:aws:iam::123456789012:group/Admins",
    "GroupName": "Admins"
  },
  "Users": []
}
```

Per ulteriori informazioni, vedere [IAM Identità \(utenti, gruppi di utenti e ruoli\)](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [GetGroup](#) in AWS CLI Command Reference.

## get-instance-profile

Il seguente esempio di codice mostra come utilizzare `get-instance-profile`.

### AWS CLI

Per ottenere informazioni sul profilo di un'istanza

Il `get-instance-profile` comando seguente ottiene informazioni sul profilo di istanza denominato `ExampleInstanceProfile`.

```
aws iam get-instance-profile \  
  --instance-profile-name ExampleInstanceProfile
```

Output:

```
{  
  "InstanceProfile": {  
    "InstanceProfileId": "AID2MAB8DPLSRHEXAMPLE",  
    "Roles": [  
      {  
        "AssumeRolePolicyDocument": "<URL-encoded-JSON>",  
        "RoleId": "AIDGPMS9R04H3FEXAMPLE",  
        "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Utilizzo dei profili di istanza](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [GetInstanceProfile](#) in AWS CLI Command Reference.

## get-login-profile

Il seguente esempio di codice mostra come utilizzare `get-login-profile`.

### AWS CLI

Per ottenere informazioni sulla password di un IAM utente

Il `get-login-profile` comando seguente consente di ottenere informazioni sulla password per l'IAM utente denominato `Bob`.

```
aws iam get-login-profile \  
  --user-name Bob
```

**Output:**

```
{
  "LoginProfile": {
    "UserName": "Bob",
    "CreateDate": "2012-09-21T23:03:39Z"
  }
}
```

Il `get-login-profile` comando può essere utilizzato per verificare che un IAM utente disponga di una password. Il comando restituisce un `NoSuchEntity` errore se non è definita alcuna password per l'utente.

Non è possibile visualizzare una password utilizzando questo comando. Se la password viene persa, è possibile reimpostare la password (`update-login-profile`) per l'utente. In alternativa, è possibile eliminare il profilo di accesso (`delete-login-profile`) per l'utente e quindi crearne uno nuovo (`create-login-profile`).

Per ulteriori informazioni, consulta [Gestione delle password per IAM gli utenti](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [GetLoginProfile](#) in AWS CLI Command Reference.

**get-mfa-device**

Il seguente esempio di codice mostra come utilizzare `get-mfa-device`.

**AWS CLI**

Per recuperare informazioni su una chiave di FIDO sicurezza

Il seguente esempio di `get-mfa-device` comando recupera informazioni sulla chiave di FIDO sicurezza specificata.

```
aws iam get-mfa-device \
  --serial-number arn:aws:iam::123456789012:u2f/user/alice/fidokeyname-EXAMPLEBN5FHTECLFG7EXAMPLE
```

**Output:**

```
{
```



Per ulteriori informazioni, consulta [Creazione di provider di identità OpenID Connect \(OIDC\) nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, consultate [GetOpenIdConnectProvider AWS CLI Command Reference](#).

## get-organizations-access-report

Il seguente esempio di codice mostra come utilizzare `get-organizations-access-report`.

### AWS CLI

Per recuperare un rapporto di accesso

L'`get-organizations-access-report` seguente visualizza un rapporto di accesso generato in precedenza per un'entità AWS Organizations. Per generare un report, utilizzare il `generate-organizations-access-report` comando.

```
aws iam get-organizations-access-report \
  --job-id a8b6c06f-aaa4-8xmp-28bc-81da71836359
```

Output:

```
{
  "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
    },
    ...
  ]
}
```

Per ulteriori informazioni, vedere [Perfezionamento delle autorizzazioni nell' AWS uso delle informazioni dell'ultimo accesso nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [GetOrganizationsAccessReport](#) in AWS CLI Command Reference.

## get-policy-version

Il seguente esempio di codice mostra come utilizzare `get-policy-version`.

### AWS CLI

Come recuperare informazioni sulla versione specificata della policy gestita specificata

Questo esempio restituisce il documento di policy per la versione v2 della policy il cui ARN è `arn:aws:iam::123456789012:policy/MyManagedPolicy`.

```
aws iam get-policy-version \  
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
  --version-id v2
```

Output:

```
{  
  "PolicyVersion": {  
    "Document": {  
      "Version": "2012-10-17",  
      "Statement": [  
        {  
          "Effect": "Allow",  
          "Action": "iam:*",  
          "Resource": "*"  
        }  
      ]  
    },  
    "VersionId": "v2",  
    "IsDefaultVersion": true,  
    "CreateDate": "2023-04-11T00:22:54+00:00"  
  }  
}
```

Per ulteriori informazioni, vedere [Politiche e autorizzazioni IAM nella Guida](#) per l'AWS IAM utente.

- Per API i dettagli, vedere [GetPolicyVersion](#) in AWS CLI Command Reference.

## get-policy

Il seguente esempio di codice mostra come utilizzare `get-policy`.



## AWS CLI

Come recuperare informazioni sulla policy gestita specificata

Questo esempio restituisce dettagli sulla politica gestita di cui ARN è `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam get-policy \  
  --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Output:

```
{  
  "Policy": {  
    "PolicyName": "MySamplePolicy",  
    "CreateDate": "2015-06-17T19:23:32Z",  
    "AttachmentCount": 0,  
    "IsAttachable": true,  
    "PolicyId": "Z27SI6FQMG2EXAMPLE1",  
    "DefaultVersionId": "v1",  
    "Path": "/",  
    "Arn": "arn:aws:iam::123456789012:policy/MySamplePolicy",  
    "UpdateDate": "2015-06-17T19:23:32Z"  
  }  
}
```

Per ulteriori informazioni, vedere [Politiche e autorizzazioni IAM nella Guida per l'AWS IAM](#) utente.

- Per API i dettagli, vedere [GetPolicy](#) in AWS CLI Command Reference.

## get-role-policy

Il seguente esempio di codice mostra come utilizzare `get-role-policy`.

### AWS CLI

Per ottenere informazioni su una politica associata a un IAM ruolo

Il `get-role-policy` comando seguente ottiene informazioni sulla politica specificata allegata al ruolo denominato `Test-Role`.

```
aws iam get-role-policy \  
  --role-name Test-Role
```

```
--role-name Test-Role \  
--policy-name ExamplePolicy
```

Output:

```
{  
  "RoleName": "Test-Role",  
  "PolicyDocument": {  
    "Statement": [  
      {  
        "Action": [  
          "s3:ListBucket",  
          "s3:Put*",  
          "s3:Get*",  
          "s3:*MultipartUpload*"  
        ],  
        "Resource": "*",  
        "Effect": "Allow",  
        "Sid": "1"  
      }  
    ]  
  }  
  "PolicyName": "ExamplePolicy"  
}
```

Per ulteriori informazioni, vedere [Creazione di IAM ruoli](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [GetRolePolicy](#) in AWS CLI Command Reference.

## get-role

Il seguente esempio di codice mostra come utilizzare `get-role`.

### AWS CLI

Per ottenere informazioni su un IAM ruolo

Il comando `get-role` seguente ottiene informazioni sul ruolo denominato `Test-Role`.

```
aws iam get-role \  
  --role-name Test-Role
```

**Output:**

```
{
  "Role": {
    "Description": "Test Role",
    "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
    "MaxSessionDuration": 3600,
    "RoleId": "AR0A1234567890EXAMPLE",
    "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"
  }
}
```

Il comando visualizza la policy di attendibilità associata al ruolo. Per elencare le policy di autorizzazioni collegate a un ruolo, usa il comando `list-role-policies`.

Per ulteriori informazioni, consulta [Creazione di IAM ruoli](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [GetRole](#) in AWS CLI Command Reference.

**get-saml-provider**

Il seguente esempio di codice mostra come utilizzare `get-saml-provider`.

**AWS CLI**

Per recuperare il metadocumento del SAML provider

Questo esempio recupera i dettagli sul provider 2.0 il cui nome è SAML. ARM  
`arn:aws:iam::123456789012:saml-provider/SAMLADFS` La risposta include il documento di metadati ricevuto dal provider di identità per creare l'entità AWS SAML provider, nonché le date di creazione e scadenza.

```
aws iam get-saml-provider \
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

**Output:**

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Creazione di provider di IAM SAML identità](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [GetSamlProvider](#) in AWS CLI Command Reference.

**get-server-certificate**

Il seguente esempio di codice mostra come utilizzare `get-server-certificate`.

**AWS CLI**

Per ottenere dettagli su un certificato server nel tuo AWS account

Il `get-server-certificate` comando seguente recupera tutti i dettagli sul certificato server specificato nel tuo AWS account.

```
aws iam get-server-certificate \
  --server-certificate-name myUpdatedServerCertificate
```

**Output:**

```
{
  "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-----
MIICiTCcAFICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWVxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
b2x1MRIwEAYDVQQDEwLUZXN0Q21sYWVxHzAdBgkqhkiG9w0BCQEWEG5vb251QGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcvcQAaRHhdLQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUHVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJIIJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvrszlaEXAMPLE=-----END CERTIFICATE-----",
  "CertificateChain": "-----BEGIN CERTIFICATE-----\nMIICiTCcAFICCCQD6md
7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMCMVVMxCzAJBgNVBAGT
AldBMRAwDgYDVQQHEwdTZWF0drGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAw
TC01BTSBDb25zb2x1MRIwEAYDVQsQQDEwLUZXN0Q21sYWVxHzAdBgkqhkiG9w0BCQ
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBh
MCMVVMxCzAJBgNVBAGTAldBMRAwDgsYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBb
WF6b24xFDASBgNVBAwTC01BTSBDb2d5zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWVx
HzAdBgkqhkiG9w0BCQEWEG5vb251QGFtYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIGWJ21uUSfwfEvySWtC2XADZ4nB+BLyGVI
k60CpiwsZ3G93vUEI03IyNoH/f0wYK8mh9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gjpEIbb30hjZnzcvcQAaRHhdLQWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCku4nUHVxYUntneD9+h8Mg9q6q+auN
KyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0F1kbFFbjvSfpJIIJ00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjS;TbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEWEG5vb251QGFtsYXpvbiEXAMPLE=\n-----END CERTIFICATE-----"
}
}

```

Per elencare i certificati server disponibili nel tuo AWS account, usa il `list-server-certificates` comando.

Per ulteriori informazioni, consulta [la sezione Gestione dei certificati del server IAM nella Guida AWS IAM per l'utente](#).

- Per API i dettagli, vedere [GetServerCertificates](#) in AWS CLI Command Reference.

## get-service-last-accessed-details-with-entities

Il seguente esempio di codice mostra come utilizzare `get-service-last-accessed-details-with-entities`.

### AWS CLI

Per recuperare un rapporto di accesso al servizio con i dettagli relativi a un servizio

L'`get-service-last-accessed-details-with-entities` esempio seguente recupera un report che contiene dettagli sugli IAM utenti e altre entità che hanno avuto accesso al servizio specificato. Per generare un report, utilizzare il `generate-service-last-accessed-details` comando. Per ottenere un elenco di servizi a cui si accede con i namespace, usa `get-service-last-accessed-details`

```
aws iam get-service-last-accessed-details-with-entities \  
  --job-id 78b6c2ba-d09e-6xmp-7039-ecde30b26916 \  
  --service-namespace Lambda
```

Output:

```
{  
  "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"
  ]
}

```

Per ulteriori informazioni, consulta [Ridefinizione delle autorizzazioni nell' AWS uso delle informazioni dell'ultimo accesso nella Guida per l'utente.AWS IAM](#)

- Per API i dettagli, vedere [GetServiceLastAccessedDetailsWithEntities](#) in AWS CLI Command Reference.

## get-service-last-accessed-details

Il seguente esempio di codice mostra come utilizzare `get-service-last-accessed-details`.

### AWS CLI

Per recuperare un rapporto di accesso al servizio

L'`get-service-last-accessed-details` esempio seguente recupera un report generato in precedenza che elenca i servizi a cui accedono le entità. IAM Per generare un report, utilizzare il `generate-service-last-accessed-details` comando.

```

aws iam get-service-last-accessed-details \
  --job-id 2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc

```

Output:

```

{
  "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
  },
]
}
```

Per ulteriori informazioni, vedere [Perfezionamento delle autorizzazioni nell' AWS uso delle informazioni dell'ultimo accesso nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [GetServiceLastAccessedDetails](#) in AWS CLI Command Reference.

## get-service-linked-role-deletion-status

Il seguente esempio di codice mostra come utilizzare `get-service-linked-role-deletion-status`.

### AWS CLI

Come verificare lo stato di una richiesta di eliminazione di un ruolo collegato a un servizio

L'esempio `get-service-linked-role-deletion-status` seguente visualizza lo stato di una precedente richiesta di eliminazione di un ruolo collegato a un servizio. L'operazione di eliminazione avviene in modo asincrono. Quando effettui la richiesta, ottieni un valore `DeletionTaskId` che hai fornito come parametro per questo comando.

```
aws iam get-service-linked-role-deletion-status \
  --deletion-task-id task/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE
```

Output:

```
{
  "Status": "SUCCEEDED"
}
```

Per ulteriori informazioni, vedere [Utilizzo dei ruoli collegati ai servizi nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [GetServiceLinkedRoleDeletionStatus](#) in AWS CLI Command Reference.



## get-ssh-public-key

Il seguente esempio di codice mostra come utilizzare `get-ssh-public-key`.

### AWS CLI

Esempio 1: recuperare una chiave SSH pubblica allegata a un IAM utente in forma SSH codificata

Il `get-ssh-public-key` comando seguente recupera la chiave SSH pubblica specificata dall'utente. IAM `sofia` L'output è in SSH codifica.

```
aws iam get-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-id APKA123456789EXAMPLE \  
  --encoding SSH
```

Output:

```
{  
  "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"  
  }  
}
```

Esempio 2: recuperare una chiave SSH pubblica allegata a un IAM utente in PEM forma codificata

Il `get-ssh-public-key` comando seguente recupera la chiave SSH pubblica specificata dall'utente. IAM `sofia` L'output è in PEM codifica.

```
aws iam get-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-id APKA123456789EXAMPLE \  
  --encoding PEM
```

Output:

```
{
```

```

    "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"
    }
  }
}

```

Per ulteriori informazioni, consulta [Use SSH keys and SSH with CodeCommit](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [GetSshPublicKey](#) in AWS CLI Command Reference.

## get-user-policy

Il seguente esempio di codice mostra come utilizzare `get-user-policy`.

### AWS CLI

Per elencare i dettagli delle politiche per un IAM utente

Il `get-user-policy` comando seguente elenca i dettagli della politica specificata allegata all'IAMutente denominato `Bob`.

```

aws iam get-user-policy \
  --user-name Bob \
  --policy-name ExamplePolicy

```

Output:

```

{
  "UserName": "Bob",
  "PolicyName": "ExamplePolicy",
  "PolicyDocument": {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Action": "*",
        "Resource": "*"
      }
    ]
  }
}

```

```
        "Effect": "Allow"
      }
    ]
  }
}
```

Per ottenere un elenco di politiche per un IAM utente, utilizzare il `list-user-policies` comando.

Per ulteriori informazioni, vedere [Politiche e autorizzazioni IAM nella Guida per l'AWS IAM utente](#).

- Per API i dettagli, vedere [GetUserPolicy](#) in AWS CLI Command Reference.

## get-user

Il seguente esempio di codice mostra come utilizzare `get-user`.

### AWS CLI

Per ottenere informazioni su un IAM utente

Il `get-user` comando seguente consente di ottenere informazioni sull'IAM utente denominato `Paulo`.

```
aws iam get-user \
  --user-name Paulo
```

Output:

```
{
  "User": {
    "UserName": "Paulo",
    "Path": "/",
    "CreateDate": "2019-09-21T23:03:13Z",
    "UserId": "AIDA123456789EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:user/Paulo"
  }
}
```

Per ulteriori informazioni, vedere [Gestione IAM degli utenti](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [GetUser](#) in AWS CLI Command Reference.

## list-access-keys

Il seguente esempio di codice mostra come utilizzare `list-access-keys`.

### AWS CLI

Per elencare la chiave di accesso IDs di un IAM utente

Il `list-access-keys` comando seguente elenca le chiavi di accesso IDs per l'IAM utente denominato `Bob`.

```
aws iam list-access-keys \  
  --user-name Bob
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Non è possibile elencare le chiavi di accesso segrete per IAM gli utenti. Se le chiavi di accesso segrete vengono perse, devi creare nuove chiavi di accesso utilizzando il comando `create-access-keys`.

Per ulteriori informazioni, vedere [Gestione delle chiavi di accesso per IAM gli utenti](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [ListAccessKeys](#) in AWS CLI Command Reference.

## list-account-aliases

Il seguente esempio di codice mostra come utilizzare `list-account-aliases`.

### AWS CLI

Elencare gli alias di un account

Il comando `list-account-aliases` seguente elenca gli alias per l'account corrente.

```
aws iam list-account-aliases
```

Output:

```
{
  "AccountAliases": [
    "mycompany"
  ]
}
```

Per ulteriori informazioni, consulta l'[ID AWS dell'account e il relativo alias](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, consulta [ListAccountAliases AWS CLI Command Reference](#).

## list-attached-group-policies

Il seguente esempio di codice mostra come utilizzare `list-attached-group-policies`.

### AWS CLI

Per elencare tutte le politiche gestite allegate al gruppo specificato

Questo esempio restituisce i nomi e ARNs le politiche gestite allegate al IAM gruppo denominato Admins nell' AWS account.

```
aws iam list-attached-group-policies \
  --group-name Admins
```

Output:

```
{
```

```
"AttachedPolicies": [  
  {  
    "PolicyName": "AdministratorAccess",  
    "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"  
  },  
  {  
    "PolicyName": "SecurityAudit",  
    "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"  
  }  
],  
"IsTruncated": false  
}
```

Per ulteriori informazioni, vedere [Politiche e autorizzazioni IAM nella Guida per l'AWS IAM utente](#).

- Per API i dettagli, vedere [ListAttachedGroupPolicies](#) in AWS CLI Command Reference.

## list-attached-role-policies

Il seguente esempio di codice mostra come utilizzare `list-attached-role-policies`.

### AWS CLI

Come elencare tutte le policy gestite collegate al ruolo specificato

Questo comando restituisce i nomi e ARNs le politiche gestite allegate al IAM ruolo denominato `SecurityAuditRole` nell' AWS account.

```
aws iam list-attached-role-policies \  
  --role-name SecurityAuditRole
```

Output:

```
{  
  "AttachedPolicies": [  
    {  
      "PolicyName": "SecurityAudit",  
      "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"  
    }  
  ],  
  "IsTruncated": false  
}
```

Per ulteriori informazioni, vedere [Politiche e autorizzazioni IAM nella Guida per l'AWS IAMutente](#).

- Per API i dettagli, vedere [ListAttachedRolePolicies](#) in AWS CLI Command Reference.

## **list-attached-user-policies**

Il seguente esempio di codice mostra come utilizzare `list-attached-user-policies`.

### AWS CLI

Per elencare tutte le politiche gestite allegate all'utente specificato

Questo comando restituisce i nomi e ARNs le politiche gestite per l'IAMutente denominato Bob nell' AWS account.

```
aws iam list-attached-user-policies \  
  --user-name Bob
```

Output:

```
{  
  "AttachedPolicies": [  
    {  
      "PolicyName": "AdministratorAccess",  
      "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"  
    },  
    {  
      "PolicyName": "SecurityAudit",  
      "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"  
    }  
  ],  
  "IsTruncated": false  
}
```

Per ulteriori informazioni, vedere [Politiche e autorizzazioni IAM nella Guida per l'AWS IAMutente](#).

- Per API i dettagli, vedere [ListAttachedUserPolicies](#) in AWS CLI Command Reference.

## **list-entities-for-policy**

Il seguente esempio di codice mostra come utilizzare `list-entities-for-policy`.

## AWS CLI

Per elencare tutti gli utenti, i gruppi e i ruoli a cui è associata la politica gestita specificata

Questo esempio restituisce un elenco di IAM gruppi, ruoli e utenti a cui è associata la politica `arn:aws:iam::123456789012:policy/TestPolicy`.

```
aws iam list-entities-for-policy \  
--policy-arn arn:aws:iam::123456789012:policy/TestPolicy
```

Output:

```
{  
  "PolicyGroups": [  
    {  
      "GroupName": "Admins",  
      "GroupId": "AGPACKCEVSQ6C2EXAMPLE"  
    }  
  ],  
  "PolicyUsers": [  
    {  
      "UserName": "Alice",  
      "UserId": "AIDACKCEVSQ6C2EXAMPLE"  
    }  
  ],  
  "PolicyRoles": [  
    {  
      "RoleName": "DevRole",  
      "RoleId": "AR0ADBQP57FF2AEXAMPLE"  
    }  
  ],  
  "IsTruncated": false  
}
```

Per ulteriori informazioni, vedere [Politiche e autorizzazioni IAM nella Guida per l'AWS IAM utente](#).

- Per API i dettagli, vedere [ListEntitiesForPolicy](#) in AWS CLI Command Reference.

### **list-group-policies**

Il seguente esempio di codice mostra come utilizzare `list-group-policies`.



## AWS CLI

Per elencare tutte le politiche in linea allegate al gruppo specificato

Il `list-group-policies` comando seguente elenca i nomi delle politiche in linea allegate al IAM gruppo denominato `Admins` nell'account corrente.

```
aws iam list-group-policies \  
  --group-name Admins
```

Output:

```
{  
  "PolicyNames": [  
    "AdminRoot",  
    "ExamplePolicy"  
  ]  
}
```

Per ulteriori informazioni, vedere [Gestione delle IAM politiche](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [ListGroupPolicies](#) in AWS CLI Command Reference.

## **list-groups-for-user**

Il seguente esempio di codice mostra come utilizzare `list-groups-for-user`.

## AWS CLI

Per elencare i gruppi a cui appartiene un IAM utente

Il `list-groups-for-user` comando seguente visualizza i gruppi a cui `Bob` appartiene l'IAM utente denominato.

```
aws iam list-groups-for-user \  
  --user-name Bob
```

Output:

```
{  
  "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"
  }
]
```

Per ulteriori informazioni, vedere [Gestione dei gruppi di IAM utenti](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [ListGroupsWithUser](#) in AWS CLI Command Reference.

## list-groups

Il seguente esempio di codice mostra come utilizzare `list-groups`.

### AWS CLI

Per elencare i IAM gruppi per l'account corrente

Il `list-groups` comando seguente elenca i IAM gruppi presenti nell'account corrente.

```
aws iam list-groups
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Gestione dei gruppi di IAM utenti](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [ListGroups](#) in AWS CLI Command Reference.

## list-instance-profile-tags

Il seguente esempio di codice mostra come utilizzare `list-instance-profile-tags`.

### AWS CLI

Per elencare i tag associati a un profilo di istanza

Il `list-instance-profile-tags` comando seguente recupera l'elenco dei tag associati al profilo di istanza specificato.

```
aws iam list-instance-profile-tags \
  --instance-profile-name deployment-role
```

Output:

```
{
  "Tags": [
    {
      "Key": "DeptID",
      "Value": "123456"
    },
    {
      "Key": "Department",
      "Value": "Accounting"
    }
  ]
}
```

```
]
}
```

Per ulteriori informazioni, vedete [Tagging IAM resources](#) nella AWS IAM User Guide.

- Per API i dettagli, vedere [ListInstanceProfileTags](#) in AWS CLI Command Reference.

## list-instance-profiles-for-role

Il seguente esempio di codice mostra come utilizzare `list-instance-profiles-for-role`.

### AWS CLI

Per elencare i profili di istanza per un IAM ruolo

Il `list-instance-profiles-for-role` comando seguente elenca i profili di istanza associati al ruolo `Test-Role`.

```
aws iam list-instance-profiles-for-role \
  --role-name Test-Role
```

Output:

```
{
  "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"
    }
  ]
}
```

```
    }
  ]
}
```

Per ulteriori informazioni, vedere [Utilizzo dei profili di istanza](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [ListInstanceProfilesForRole](#) in AWS CLI Command Reference.

## list-instance-profiles

Il seguente esempio di codice mostra come utilizzare `list-instance-profiles`.

### AWS CLI

Per elencare i profili di istanza per l'account

Il `list-instance-profiles` comando seguente elenca i profili di istanza associati all'account corrente.

```
aws iam list-instance-profiles
```

Output:

```
{
  "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",
```



## list-mfa-device-tags

Il seguente esempio di codice mostra come utilizzare `list-mfa-device-tags`.

### AWS CLI

Per elencare i tag allegati a un MFA dispositivo

Il `list-mfa-device-tags` comando seguente recupera l'elenco dei tag associati al MFA dispositivo specificato.

```
aws iam list-mfa-device-tags \
  --serial-number arn:aws:iam::123456789012:mfa/alice
```

Output:

```
{
  "Tags": [
    {
      "Key": "DeptID",
      "Value": "123456"
    },
    {
      "Key": "Department",
      "Value": "Accounting"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Taggare IAM le risorse nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [ListMfaDeviceTags](#) in AWS CLI Command Reference.

## list-mfa-devices

Il seguente esempio di codice mostra come utilizzare `list-mfa-devices`.

### AWS CLI

Per elencare tutti MFA i dispositivi per un utente specificato

Questo esempio restituisce dettagli sul MFA dispositivo assegnato all'IAMutenteBob.

```
aws iam list-mfa-devices \  
  --user-name Bob
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedete [Using Multi-Factor Authentication \(MFA\) AWS nella Guida per l'AWS IAM](#) utente.

- Per API i dettagli, vedere [ListMfaDevices](#) in AWS CLI Command Reference.

## list-open-id-connect-provider-tags

Il seguente esempio di codice mostra come utilizzare `list-open-id-connect-provider-tags`.



## AWS CLI

Per elencare i tag allegati a un provider di identità compatibile con OpenID Connect (OIDC)

Il `list-open-id-connect-provider-tags` comando seguente recupera l'elenco dei tag associati al provider di identità specificato OIDC.

```
aws iam list-open-id-connect-provider-tags \
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

Output:

```
{
  "Tags": [
    {
      "Key": "DeptID",
      "Value": "123456"
    },
    {
      "Key": "Department",
      "Value": "Accounting"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Taggare IAM le risorse nella Guida](#) per l'AWS IAM utente.

- Per API i dettagli, vedere [ListOpenIdConnectProviderTags](#) in AWS CLI Command Reference.

## **list-open-id-connect-providers**

Il seguente esempio di codice mostra come utilizzare `list-open-id-connect-providers`.

### AWS CLI

Per elencare informazioni sui provider OpenID Connect presenti nell'account AWS

Questo esempio restituisce un elenco ARNS di tutti i provider OpenID Connect definiti nell'account corrente AWS .

```
aws iam list-open-id-connect-providers
```

**Output:**

```
{
  "OpenIDConnectProviderList": [
    {
      "Arn": "arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Creazione di provider di identità OpenID Connect \(OIDC\) nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, consultate [ListOpenIdConnectProviders AWS CLI](#) Command Reference.

**list-policies-granting-service-access**

Il seguente esempio di codice mostra come utilizzare `list-policies-granting-service-access`.

**AWS CLI**

Per elencare le politiche che garantiscono l'accesso principale al servizio specificato

L'`list-policies-granting-service-access`esempio seguente recupera l'elenco delle politiche che concedono IAM all'utente l'`sofia`accesso al AWS CodeCommit servizio.

```
aws iam list-policies-granting-service-access \
  --arn arn:aws:iam::123456789012:user/sofia \
  --service-namespaces codecommit
```

**Output:**

```
{
  "PoliciesGrantingServiceAccess": [
    {
      "ServiceNamespace": "codecommit",
      "Policies": [
        {
          "PolicyName": "Grant-Sofia-Access-To-CodeCommit",
```

```
        "PolicyType": "INLINE",
        "EntityType": "USER",
        "EntityName": "sofia"
      }
    ]
  },
  "IsTruncated": false
}
```

Per ulteriori informazioni, consulta [Utilizzare IAM con CodeCommit: credenziali Git, SSH chiavi e chiavi di AWS accesso](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [ListPoliciesGrantingServiceAccess](#) in AWS CLI Command Reference.

## list-policies

Il seguente esempio di codice mostra come utilizzare `list-policies`.

### AWS CLI

Per elencare le politiche gestite disponibili per il tuo AWS account

Questo esempio restituisce una raccolta delle prime due politiche gestite disponibili nell'AWS account corrente.

```
aws iam list-policies \
  --max-items 3
```

Output:

```
{
  "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": "ANPAJYRXTHIB4FOVS3ZXS",
    "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=="
}

```

Per ulteriori informazioni, consulta la sezione [Politiche e autorizzazioni IAM nella Guida per l'AWS IAMutente](#).

- Per API i dettagli, vedere [ListPolicies](#) in AWS CLI Command Reference.

## list-policy-tags

Il seguente esempio di codice mostra come utilizzare `list-policy-tags`.

### AWS CLI

Per elencare i tag allegati a una politica gestita

Il `list-policy-tags` comando seguente recupera l'elenco dei tag associati alla politica gestita specificata.

```
aws iam list-policy-tags \  
  --policy-arn arn:aws:iam::123456789012:policy/billing-access
```

Output:

```
{  
  "Tags": [  
    {  
      "Key": "DeptID",  
      "Value": "123456"  
    },  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Tagging IAM resources](#) nella AWS IAM User Guide.

- Per API i dettagli, vedere [ListPolicyTags](#) in AWS CLI Command Reference.

## **list-policy-versions**

Il seguente esempio di codice mostra come utilizzare `list-policy-versions`.

AWS CLI

Per elencare informazioni sulle versioni della politica gestita specificata

Questo esempio restituisce l'elenco delle versioni disponibili della politica la cui ARN è `arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam list-policy-versions \  
  --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

Output:

```
{
  "IsTruncated": false,
  "Versions": [
    {
      "VersionId": "v2",
      "IsDefaultVersion": true,
      "CreateDate": "2015-06-02T23:19:44Z"
    },
    {
      "VersionId": "v1",
      "IsDefaultVersion": false,
      "CreateDate": "2015-06-02T22:30:47Z"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Politiche e autorizzazioni IAM nella Guida per l'AWS IAM](#) utente.

- Per API i dettagli, vedere [ListPolicyVersions](#) in AWS CLI Command Reference.

## list-role-policies

Il seguente esempio di codice mostra come utilizzare `list-role-policies`.

### AWS CLI

Per elencare le politiche associate a un IAM ruolo

Il `list-role-policies` comando seguente elenca i nomi delle politiche di autorizzazione per il IAM ruolo specificato.

```
aws iam list-role-policies \
  --role-name Test-Role
```

Output:

```
{
  "PolicyNames": [
    "ExamplePolicy"
  ]
}
```

Per consultare la policy di attendibilità collegata a un ruolo, usa il comando `get-role`. Per visualizzare i dettagli di una policy di autorizzazioni, usa il comando `get-role-policy`.

Per ulteriori informazioni, vedere [Creazione di IAM ruoli](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [ListRolePolicies](#) in AWS CLI Command Reference.

## **list-role-tags**

Il seguente esempio di codice mostra come utilizzare `list-role-tags`.

### AWS CLI

Per elencare i tag associati a un ruolo

Il `list-role-tags` comando seguente recupera l'elenco dei tag associati al ruolo specificato.

```
aws iam list-role-tags \  
  --role-name production-role
```

Output:

```
{  
  "Tags": [  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    },  
    {  
      "Key": "DeptID",  
      "Value": "12345"  
    }  
  ],  
  "IsTruncated": false  
}
```

Per ulteriori informazioni, consulta [Taggare IAM le risorse nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [ListRoleTags](#) in AWS CLI Command Reference.

## **list-roles**

Il seguente esempio di codice mostra come utilizzare `list-roles`.

## AWS CLI

Per elencare IAM i ruoli per l'account corrente

Il `list-roles` comando seguente elenca IAM i ruoli per l'account corrente.

```
aws iam list-roles
```

Output:

```
{
  "Roles": [
    {
      "Path": "/",
      "RoleName": "ExampleRole",
      "RoleId": "AROAJ520TH4H7LEXAMPLE",
      "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
}
]
}

```

Per ulteriori informazioni, vedere [Creazione di IAM ruoli](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [ListRoles](#) in AWS CLI Command Reference.

## list-saml-provider-tags

Il seguente esempio di codice mostra come utilizzare `list-saml-provider-tags`.

### AWS CLI

Per elencare i tag associati a un SAML provider

Il `list-saml-provider-tags` comando seguente recupera l'elenco dei tag associati al SAML provider specificato.

```

aws iam list-saml-provider-tags \
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS

```

Output:

```

{
  "Tags": [
    {
      "Key": "DeptID",
      "Value": "123456"
    },
    {

```

```
        "Key": "Department",
        "Value": "Accounting"
    }
]
}
```

Per ulteriori informazioni, consulta [Taggare IAM le risorse nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [ListSamlProviderTags](#) in AWS CLI Command Reference.

## list-saml-providers

Il seguente esempio di codice mostra come utilizzare `list-saml-providers`.

### AWS CLI

Per elencare i SAML provider presenti nell' AWS account

Questo esempio recupera l'elenco dei provider SAML 2.0 creati nell' AWS account corrente.

```
aws iam list-saml-providers
```

Output:

```
{
  "SAMLProviderList": [
    {
      "Arn": "arn:aws:iam::123456789012:saml-provider/SAML-ADFS",
      "ValidUntil": "2015-06-05T22:45:14Z",
      "CreateDate": "2015-06-05T22:45:14Z"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Creazione di provider di IAM SAML identità](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [ListSAMLProviders](#) in AWS CLI Command Reference.

## list-server-certificate-tags

Il seguente esempio di codice mostra come utilizzare `list-server-certificate-tags`.

## AWS CLI

Per elencare i tag allegati a un certificato del server

Il `list-server-certificate-tags` comando seguente recupera l'elenco dei tag associati al certificato del server specificato.

```
aws iam list-server-certificate-tags \  
  --server-certificate-name ExampleCertificate
```

Output:

```
{  
  "Tags": [  
    {  
      "Key": "DeptID",  
      "Value": "123456"  
    },  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Tagging IAM resources](#) nella AWS IAMUser Guide.

- Per API i dettagli, vedere [ListServerCertificateTags](#) in AWS CLI Command Reference.

## **list-server-certificates**

Il seguente esempio di codice mostra come utilizzare `list-server-certificates`.

## AWS CLI

Per elencare i certificati del server presenti nel tuo AWS account

Il `list-server-certificates` comando seguente elenca tutti i certificati server archiviati e disponibili per l'uso nell' AWS account.

```
aws iam list-server-certificates
```

**Output:**

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [la sezione Gestione dei certificati del server IAM nella Guida AWS IAM per l'utente](#).

- Per API i dettagli, vedere [ListServerCertificates](#) in AWS CLI Command Reference.

**list-service-specific-credential**

Il seguente esempio di codice mostra come utilizzare `list-service-specific-credential`.

**AWS CLI**

Esempio 1: Elenca le credenziali specifiche del servizio per un utente

L'`list-service-specific-credential` esempio seguente visualizza tutte le credenziali specifiche del servizio assegnate all'utente specificato. Le password non sono incluse nella risposta.

```
aws iam list-service-specific-credentials \
```

```
--user-name sofia
```

Output:

```
{
  "ServiceSpecificCredential": {
    "CreateDate": "2019-04-18T20:45:36+00:00",
    "ServiceName": "codecommit.amazonaws.com",
    "ServiceUserName": "sofia-at-123456789012",
    "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
    "UserName": "sofia",
    "Status": "Active"
  }
}
```

Esempio 2: Elenca le credenziali specifiche del servizio per un utente filtrato in base a un servizio specifico

L'`list-service-specific-credentials` esempio seguente visualizza le credenziali specifiche del servizio assegnate all'utente che effettua la richiesta. L'elenco viene filtrato per includere solo le credenziali per il servizio specificato. Le password non sono incluse nella risposta.

```
aws iam list-service-specific-credentials \
  --service-name codecommit.amazonaws.com
```

Output:

```
{
  "ServiceSpecificCredential": {
    "CreateDate": "2019-04-18T20:45:36+00:00",
    "ServiceName": "codecommit.amazonaws.com",
    "ServiceUserName": "sofia-at-123456789012",
    "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
    "UserName": "sofia",
    "Status": "Active"
  }
}
```

Per ulteriori informazioni, consulta [Create Git credenziali per HTTPS le connessioni CodeCommit](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [ListServiceSpecificCredential](#) in AWS CLI Command Reference.

## list-service-specific-credentials

Il seguente esempio di codice mostra come utilizzare `list-service-specific-credentials`.

### AWS CLI

Per recuperare un elenco di credenziali

L'`list-service-specific-credentials` seguente elenca le credenziali generate per HTTPS l'accesso ai AWS CodeCommit repository per un utente denominato. `developer`

```
aws iam list-service-specific-credentials \  
  --user-name developer \  
  --service-name codecommit.amazonaws.com
```

Output:

```
{  
  "ServiceSpecificCredentials": [  
    {  
      "UserName": "developer",  
      "Status": "Inactive",  
      "ServiceUserName": "developer-at-123456789012",  
      "CreateDate": "2019-10-01T04:31:41Z",  
      "ServiceSpecificCredentialId": "ACCAQF0DXMPL4YFHP7DZE",  
      "ServiceName": "codecommit.amazonaws.com"  
    },  
    {  
      "UserName": "developer",  
      "Status": "Active",  
      "ServiceUserName": "developer+1-at-123456789012",  
      "CreateDate": "2019-10-01T04:31:45Z",  
      "ServiceSpecificCredentialId": "ACCAQF0XMPL6VW57M7AJP",  
      "ServiceName": "codecommit.amazonaws.com"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Create Git credenziali per HTTPS le connessioni CodeCommit](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [ListServiceSpecificCredentials](#) in AWS CLI Command Reference.

## list-signing-certificates

Il seguente esempio di codice mostra come utilizzare `list-signing-certificates`.

### AWS CLI

Per elencare i certificati di firma per un IAM utente

Il `list-signing-certificates` comando seguente elenca i certificati di firma per l'IAM utente denominato `Bob`.

```
aws iam list-signing-certificates \
  --user-name Bob
```

Output:

```
{
  "Certificates": [
    {
      "UserName": "Bob",
      "Status": "Inactive",
      "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----
END CERTIFICATE-----",
      "CertificateId": "TA7SMP42TDN5Z260BPJE7EXAMPLE",
      "UploadDate": "2013-06-06T21:40:08Z"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Gestire i certificati di firma](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [ListSigningCertificates AWS CLI](#) Command Reference.

## list-ssh-public-keys

Il seguente esempio di codice mostra come utilizzare `list-ssh-public-keys`.

### AWS CLI

Per elencare le chiavi SSH pubbliche allegate a un IAM utente

L'`list-ssh-public-keys` seguente elenca le chiavi SSH pubbliche allegate all'IAM utente `sofia`.

```
aws iam list-ssh-public-keys \  
  --user-name sofia
```

Output:

```
{  
  "SSHPublicKeys": [  
    {  
      "UserName": "sofia",  
      "SSHPublicKeyId": "APKA1234567890EXAMPLE",  
      "Status": "Inactive",  
      "UploadDate": "2019-04-18T17:04:49+00:00"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Use SSH keys and SSH with CodeCommit](#) nella Guida AWS IAM per l'utente

- Per API i dettagli, vedere [ListSshPublicKeys](#) in AWS CLI Command Reference.

## `list-user-policies`

Il seguente esempio di codice mostra come utilizzare `list-user-policies`.

AWS CLI

Per elencare le politiche per un IAM utente

Il `list-user-policies` comando seguente elenca le politiche allegate all'IAM utente denominato `Bob`.

```
aws iam list-user-policies \  
  --user-name Bob
```

Output:

```
{
```



```
    "PolicyNames": [  
        "ExamplePolicy",  
        "TestPolicy"  
    ]  
}
```

Per ulteriori informazioni, consulta [Creazione di un IAM utente nel tuo AWS account](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, consulta [ListUserPolicies AWS CLI](#) Command Reference.

## list-user-tags

Il seguente esempio di codice mostra come utilizzare `list-user-tags`.

### AWS CLI

Per elencare i tag associati a un utente

Il `list-user-tags` comando seguente recupera l'elenco dei tag associati all'IAM utente specificato.

```
aws iam list-user-tags \  
    --user-name alice
```

Output:

```
{  
  "Tags": [  
    {  
      "Key": "Department",  
      "Value": "Accounting"  
    },  
    {  
      "Key": "DeptID",  
      "Value": "12345"  
    }  
  ],  
  "IsTruncated": false  
}
```

Per ulteriori informazioni, consulta [Taggare IAM le risorse nella Guida](#) per l'IAM utente.

- Per API i dettagli, vedere [ListUserTags](#) in AWS CLI Command Reference.

## list-users

Il seguente esempio di codice mostra come utilizzare `list-users`.

### AWS CLI

Per elencare IAM gli utenti

Il `list-users` comando seguente elenca gli IAM utenti dell'account corrente.

```
aws iam list-users
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Elencare IAM gli utenti](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, consulta [ListUsers AWS CLI](#) Command Reference.

## list-virtual-mfa-devices

Il seguente esempio di codice mostra come utilizzare `list-virtual-mfa-devices`.

## AWS CLI

Per elencare MFA i dispositivi virtuali

Il `list-virtual-mfa-devices` comando seguente elenca i MFA dispositivi virtuali che sono stati configurati per l'account corrente.

```
aws iam list-virtual-mfa-devices
```

Output:

```
{
  "VirtualMFADevices": [
    {
      "SerialNumber": "arn:aws:iam::123456789012:mfa/ExampleMFADevice"
    },
    {
      "SerialNumber": "arn:aws:iam::123456789012:mfa/Fred"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Abilitazione di un dispositivo virtuale di autenticazione a più fattori \(MFA\)](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [ListVirtualMfaDevices](#) in AWS CLI Command Reference.

## put-group-policy

Il seguente esempio di codice mostra come utilizzare `put-group-policy`.

## AWS CLI

Come aggiungere una policy a un gruppo

Il `put-group-policy` comando seguente aggiunge una politica al IAM gruppo denominato `Admins`.

```
aws iam put-group-policy \
  --group-name Admins \
  --policy-document file://AdminPolicy.json \
```

```
--policy-name AdminRoot
```

Questo comando non produce alcun output.

La policy è definita come JSON documento nel AdminPolicyfile.json. (Il nome e l'estensione del file non hanno importanza.)

Per ulteriori informazioni, vedere [Gestione delle IAM politiche nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [PutGroupPolicy](#) in AWS CLI Command Reference.

## put-role-permissions-boundary

Il seguente esempio di codice mostra come utilizzare `put-role-permissions-boundary`.

### AWS CLI

Esempio 1: applicare a un ruolo un limite di autorizzazioni basato su una politica personalizzata IAM

L'`put-role-permissions-boundary`esempio seguente applica la politica personalizzata `intern-boundary` denominata limite delle autorizzazioni per il ruolo specificato. IAM

```
aws iam put-role-permissions-boundary \  
  --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \  
  --role-name lambda-application-role
```

Questo comando non produce alcun output.

Esempio 2: applicare a un ruolo un limite di autorizzazioni basato su una politica AWS gestita IAM

L'`put-role-permissions-boundary`esempio seguente applica la `PowerUserAccess` politica AWS gestita come limite delle autorizzazioni per il ruolo specificato. IAM

```
aws iam put-role-permissions-boundary \  
  --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \  
  --role-name x-account-admin
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Modifica di un ruolo](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [PutRolePermissionsBoundary](#) in AWS CLI Command Reference.

## put-role-policy

Il seguente esempio di codice mostra come utilizzare `put-role-policy`.

### AWS CLI

Per allegare una politica di autorizzazioni a un ruolo IAM

Il comando `put-role-policy` seguente aggiunge una policy di autorizzazioni al ruolo denominato `Test-Role`.

```
aws iam put-role-policy \  
  --role-name Test-Role \  
  --policy-name ExamplePolicy \  
  --policy-document file://AdminPolicy.json
```

Questo comando non produce alcun output.

La policy è definita come JSON documento nel `AdminPolicyfile.json`. (Il nome e l'estensione del file non hanno importanza.)

Per collegare una policy di attendibilità a un ruolo, usa il comando `update-assume-role-policy`.

Per ulteriori informazioni, consulta [Modificare un ruolo](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [PutRolePolicy](#) in AWS CLI Command Reference.

## put-user-permissions-boundary

Il seguente esempio di codice mostra come utilizzare `put-user-permissions-boundary`.

### AWS CLI

Esempio 1: applicare a un utente un limite di autorizzazioni basato su una politica personalizzata IAM

L'`put-user-permissions-boundary` esempio seguente applica una politica personalizzata `intern-boundary` denominata limite delle autorizzazioni per l'utente specificato. IAM

```
aws iam put-user-permissions-boundary \  
  --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \  
  --user-name intern
```

Questo comando non produce alcun output.

Esempio 2: applicare a un utente un limite di autorizzazioni basato su una politica AWS gestita IAM

L'`put-user-permissions-boundary` seguente applica la policy AWS gestita `PowerUserAccess` denominata limite delle autorizzazioni per l'utente specificato. IAM

```
aws iam put-user-permissions-boundary \  
  --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \  
  --user-name developer
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Aggiungere e rimuovere le autorizzazioni di IAM identità](#) nella Guida per l'utente.AWS IAM

- Per API i dettagli, vedere [PutUserPermissionsBoundary](#) in AWS CLI Command Reference.

## **put-user-policy**

Il seguente esempio di codice mostra come utilizzare `put-user-policy`.

### AWS CLI

Per allegare una politica a un IAM utente

Il `put-user-policy` comando seguente allega una politica all'IAM utente denominato `Bob`.

```
aws iam put-user-policy \  
  --user-name Bob \  
  --policy-name ExamplePolicy \  
  --policy-document file://AdminPolicy.json
```

Questo comando non produce alcun output.

La policy è definita come JSON documento nel `AdminPolicyfile.json`. (Il nome e l'estensione del file non hanno importanza.)

Per ulteriori informazioni, consulta [Aggiungere e rimuovere le autorizzazioni di IAM identità](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [PutUserPolicy](#) in AWS CLI Command Reference.

## remove-client-id-from-open-id-connect-provider

Il seguente esempio di codice mostra come utilizzare `remove-client-id-from-open-id-connect-provider`.

### AWS CLI

Per rimuovere l'ID client specificato dall'elenco dei client IDs registrati per il provider IAM OpenID Connect specificato

Questo esempio rimuove l'ID client `My-TestApp-3` dall'elenco dei client IDs associati al IAM OIDC provider il cui nome ARN è `arn: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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creazione di provider di identità OpenID Connect \(OIDC\) nella Guida](#) per l'AWS IAM utente.

- Per API i dettagli, consultate [RemoveClientIdFromOpenIdConnectProvider AWS CLI Command Reference](#).

## remove-role-from-instance-profile

Il seguente esempio di codice mostra come utilizzare `remove-role-from-instance-profile`.

### AWS CLI

Per rimuovere un ruolo da un profilo di istanza

Il `remove-role-from-instance-profile` comando seguente rimuove il ruolo denominato `Test-Role` dal profilo di istanza denominato `ExampleInstanceProfile`.

```
aws iam remove-role-from-instance-profile \
  --instance-profile-name ExampleInstanceProfile \
  --role-name Test-Role
```

Per ulteriori informazioni, vedere [Utilizzo dei profili di istanza](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [RemoveRoleFromInstanceProfile](#) in AWS CLI Command Reference.

## **remove-user-from-group**

Il seguente esempio di codice mostra come utilizzare `remove-user-from-group`.

### AWS CLI

Per rimuovere un utente da un IAM gruppo

Il `remove-user-from-group` comando seguente rimuove l'utente denominato Bob dal IAM gruppo denominato Admins.

```
aws iam remove-user-from-group \  
  --user-name Bob \  
  --group-name Admins
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Aggiungere e rimuovere utenti in un gruppo di IAM utenti](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [RemoveUserFromGroup](#) in AWS CLI Command Reference.

## **reset-service-specific-credential**

Il seguente esempio di codice mostra come utilizzare `reset-service-specific-credential`.

### AWS CLI

Esempio 1: reimpostazione della password per una credenziale specifica del servizio allegata all'utente che effettua la richiesta

L'esempio seguente genera una nuova password crittograficamente sicura per la credenziale specifica del servizio specificata allegata all'utente che effettua la richiesta.

```
aws iam reset-service-specific-credential \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```



## Output:

```
{
  "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"
  }
}
```

Esempio 2: reimpostazione della password per una credenziale specifica del servizio allegata a un utente specificato

L'`reset-service-specific-credential` seguente genera una nuova password crittograficamente sicura per una credenziale specifica del servizio allegata all'utente specificato.

```
aws iam reset-service-specific-credential \
  --user-name sofia \
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

## Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Create Git credenziali per HTTPS le connessioni CodeCommit](#) nella Guida per l'AWS CodeCommit utente.

- Per API i dettagli, vedere [ResetServiceSpecificCredential](#) in AWS CLI Command Reference.

## resync-mfa-device

Il seguente esempio di codice mostra come utilizzare `resync-mfa-device`.

### AWS CLI

Per sincronizzare un dispositivo MFA

L'esempio seguente sincronizza il MFA dispositivo associato all'IAM utente Bob e il cui dispositivo ARN è `arn:aws:iam::123456789012:mfa/BobsMFADevice` con un programma di autenticazione che ha fornito i due codici di autenticazione.

```
aws iam resync-mfa-device \  
  --user-name Bob \  
  --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \  
  --authentication-code1 123456 \  
  --authentication-code2 987654
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Using Multi-Factor Authentication \(MFA\) AWS nella Guida per l'utente AWS IAM](#)

- Per API i dettagli, vedere [ResyncMfaDevice](#) in AWS CLI Command Reference.

## set-default-policy-version

Il seguente esempio di codice mostra come utilizzare `set-default-policy-version`.

### AWS CLI

Per impostare la versione specificata della politica specificata come versione predefinita della politica.

Questo esempio imposta la v2 versione della politica ARN utilizzata `arn:aws:iam::123456789012:policy/MyPolicy` come versione attiva predefinita.

```
aws iam set-default-policy-version \  
  --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \  
  --version-id v2
```

Per ulteriori informazioni, vedere [Politiche e autorizzazioni IAM nella Guida per l'AWS IAM utente](#).

- Per API i dettagli, vedere [SetDefaultPolicyVersion](#) in AWS CLI Command Reference.

## set-security-token-service-preferences

Il seguente esempio di codice mostra come utilizzare `set-security-token-service-preferences`.

### AWS CLI

Per impostare la versione globale del token endpoint

L'esempio seguente configura Amazon STS per utilizzare i token della versione 2 quando esegui l'autenticazione sull'endpoint globale.

```
aws iam set-security-token-service-preferences \  
  --global-endpoint-token-version v2Token
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing AWS STS in an AWS Region nella Guida](#) per l'utente AWS IAM

- Per API i dettagli, vedere [SetSecurityTokenServicePreferences](#) in AWS CLI Command Reference.

## simulate-custom-policy

Il seguente esempio di codice mostra come utilizzare `simulate-custom-policy`.

### AWS CLI

Esempio 1: simulare gli effetti di tutte le IAM politiche associate a un IAM utente o a un ruolo

Di seguito `simulate-custom-policy` viene illustrato come fornire sia la policy che definire i valori delle variabili e simulare una API chiamata per vedere se è consentita o rifiutata. L'esempio seguente mostra una politica che consente l'accesso al database solo dopo una data e un'ora specificate. La simulazione ha esito positivo perché le azioni simulate e la `aws:CurrentTime` variabile specificata soddisfano tutte i requisiti della policy.

```
aws iam simulate-custom-policy \  
  --policy-name ExamplePolicy \  
  --policy-document '{...}' \  
  --action aws:iam:CreateUser \  
  --variable aws:CurrentTime '2017-01-01T00:00:00Z'
```

```

--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

```

Output:

```

{
  "EvaluationResults": [
    {
      "EvalActionName": "dynamodb>CreateBackup",
      "EvalResourceName": "*",
      "EvalDecision": "allowed",
      "MatchedStatements": [
        {
          "SourcePolicyId": "PolicyInputList.1",
          "StartPosition": {
            "Line": 1,
            "Column": 38
          },
          "EndPosition": {
            "Line": 1,
            "Column": 167
          }
        }
      ],
      "MissingContextValues": []
    }
  ]
}

```

Esempio 2: simulare un comando proibito dalla policy

L'simulate-custom-policyesempio seguente mostra i risultati della simulazione di un comando proibito dalla policy. In questo esempio, la data fornita è precedente a quella richiesta dalla condizione della politica.

```

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"

```

Output:

```

{
  "EvaluationResults": [
    {
      "EvalActionName": "dynamodb:CreateBackup",
      "EvalResourceName": "*",
      "EvalDecision": "implicitDeny",
      "MatchedStatements": [],
      "MissingContextValues": []
    }
  ]
}

```

Per ulteriori informazioni, consulta [Testare IAM le politiche con il simulatore di IAM policy](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, consulta [SimulateCustomPolicy AWS CLI Command Reference](#).

## simulate-principal-policy

Il seguente esempio di codice mostra come utilizzare `simulate-principal-policy`.

AWS CLI

Esempio 1: simulare gli effetti di una politica arbitraria IAM

Di seguito `simulate-principal-policy` viene illustrato come simulare un utente che richiama un'API azione e determina se le politiche associate a tale utente consentono o negano l'azione. Nell'esempio seguente, l'utente dispone di una politica che consente solo l'azione `codecommit:ListRepositories`.

```

aws iam simulate-principal-policy \
--policy-source-arn arn:aws:iam::123456789012:user/alejandro \
--action-names codecommit:ListRepositories

```

Output:

```

{
  "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": []
    }
  ]
}

```

### Esempio 2: simulare gli effetti di un comando proibito

L'`simulate-custom-policy` esempio seguente mostra i risultati della simulazione di un comando proibito da una delle politiche dell'utente. Nell'esempio seguente, l'utente dispone di una policy che consente l'accesso a un database DynamoDB solo dopo una certa data e ora. La simulazione prevede che l'utente tenti di accedere al database con un `aws:CurrentTime` valore precedente a quello consentito dalle condizioni della policy.

```

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

```

Output:

```

{
  "EvaluationResults": [

```

```
{
  "EvalActionName": "dynamodb:CreateBackup",
  "EvalResourceName": "*",
  "EvalDecision": "implicitDeny",
  "MatchedStatements": [],
  "MissingContextValues": []
}
]
```

Per ulteriori informazioni, consulta [Testing IAM policies with the IAM policy simulator](#) nella Guida per l'utente.AWS IAM

- Per API i dettagli, consulta [SimulatePrincipalPolicy AWS CLI](#) Command Reference.

## tag-instance-profile

Il seguente esempio di codice mostra come utilizzare `tag-instance-profile`.

### AWS CLI

Per aggiungere un tag a un profilo di istanza

Il `tag-instance-profile` comando seguente aggiunge un tag con un nome di reparto al profilo di istanza specificato.

```
aws iam tag-instance-profile \
  --instance-profile-name deployment-role \
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [TagInstanceProfile](#)in AWS CLI Command Reference.

## tag-mfa-device

Il seguente esempio di codice mostra come utilizzare `tag-mfa-device`.

### AWS CLI

Per aggiungere un tag a un MFA dispositivo

Il `tag-mfa-device` comando seguente aggiunge un tag con un nome di reparto al MFA dispositivo specificato.

```
aws iam tag-mfa-device \  
  --serial-number arn:aws:iam::123456789012:mfa/alice \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [TagMfaDevice](#) in AWS CLI Command Reference.

## tag-open-id-connect-provider

Il seguente esempio di codice mostra come utilizzare `tag-open-id-connect-provider`.

### AWS CLI

Per aggiungere un tag a un provider di identità compatibile con OpenID Connect (OIDC)

Il `tag-open-id-connect-provider` comando seguente aggiunge un tag con un nome di reparto al provider di OIDC identità specificato.

```
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"}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [TagOpenIdConnectProvider](#) in AWS CLI Command Reference.

## tag-policy

Il seguente esempio di codice mostra come utilizzare `tag-policy`.

### AWS CLI

Per aggiungere un tag a una politica gestita dai clienti



Il `tag-policy` comando seguente aggiunge un tag con un nome di reparto alla politica gestita dai clienti specificata.

```
aws iam tag-policy \  
  --policy-arn arn:aws:iam::123456789012:policy/billing-access \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [TagPolicy](#) in AWS CLI Command Reference.

## tag-role

Il seguente esempio di codice mostra come utilizzare `tag-role`.

### AWS CLI

Per aggiungere un tag a un ruolo

Il `tag-role` comando seguente aggiunge un tag con un nome di reparto al ruolo specificato.

```
aws iam tag-role --role-name my-role \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [TagRole](#) in AWS CLI Command Reference.

## tag-saml-provider

Il seguente esempio di codice mostra come utilizzare `tag-saml-provider`.

### AWS CLI

Per aggiungere un tag a un SAML provider

Il `tag-saml-provider` comando seguente aggiunge un tag con un nome di reparto al SAML provider specificato.

```
aws iam tag-saml-provider \  
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [TagSamlProvider](#) in AWS CLI Command Reference.

## tag-server-certificate

Il seguente esempio di codice mostra come utilizzare `tag-server-certificate`.

### AWS CLI

Per aggiungere un tag a un certificato del server

Il `tag-saml-provider` comando seguente aggiunge un tag con un nome di dipartimento al certificato del server specificato.

```
aws iam tag-server-certificate \  
  --server-certificate-name ExampleCertificate \  
  --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse nella Guida](#) per l'AWS IAM utente.

- Per API i dettagli, vedere [TagServerCertificate](#) in AWS CLI Command Reference.

## tag-user

Il seguente esempio di codice mostra come utilizzare `tag-user`.

### AWS CLI

Per aggiungere un tag a un utente

Il `tag-user` comando seguente aggiunge un tag con il Dipartimento associato all'utente specificato.

```
aws iam tag-user \  
  --user-name alice \  
  --tags '{"Key": "Department", "Value": "Accounting"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [TagUser](#) in AWS CLI Command Reference.

## untag-instance-profile

Il seguente esempio di codice mostra come utilizzare `untag-instance-profile`.

### AWS CLI

Per rimuovere un tag dal profilo di un'istanza

Il `untag-instance-profile` comando seguente rimuove qualsiasi tag con il nome chiave 'Department' dal profilo di istanza specificato.

```
aws iam untag-instance-profile \  
  --instance-profile-name deployment-role \  
  --tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [UntagInstanceProfile](#) in AWS CLI Command Reference.

## untag-mfa-device

Il seguente esempio di codice mostra come utilizzare `untag-mfa-device`.

### AWS CLI

Per rimuovere un tag da un MFA dispositivo

Il `untag-mfa-device` comando seguente rimuove qualsiasi tag con il nome chiave 'Department' dal MFA dispositivo specificato.

```
aws iam untag-mfa-device \  
  --serial-number arn:aws:iam::123456789012:mfa/alice \  
  --tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [UntagMfaDevice](#) in AWS CLI Command Reference.

## untag-open-id-connect-provider

Il seguente esempio di codice mostra come utilizzare `untag-open-id-connect-provider`.

### AWS CLI

Per rimuovere un tag da un provider di OIDC identità

Il `untag-open-id-connect-provider` comando seguente rimuove qualsiasi tag con il nome chiave 'Department' dal provider di OIDC identità specificato.

```
aws iam untag-open-id-connect-provider \  
  --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/  
server.example.com \  
  --tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [UntagOpenIdConnectProvider](#) in AWS CLI Command Reference.

## untag-policy

Il seguente esempio di codice mostra come utilizzare `untag-policy`.

### AWS CLI

Per rimuovere un tag da una politica gestita dai clienti

Il `untag-policy` comando seguente rimuove qualsiasi tag con il nome chiave «Dipartimento» dalla politica gestita dai clienti specificata.

```
aws iam untag-policy \  
  --policy-arn arn:aws:iam::452925170507:policy/billing-access \  
  --tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [UntagPolicy](#) in AWS CLI Command Reference.

## untag-role

Il seguente esempio di codice mostra come utilizzare `untag-role`.

### AWS CLI

Per rimuovere un tag da un ruolo

Il `untag-role` comando seguente rimuove qualsiasi tag con il nome chiave 'Department' dal ruolo specificato.

```
aws iam untag-role \  
  --role-name my-role \  
  --tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [UntagRole](#) in AWS CLI Command Reference.

## untag-saml-provider

Il seguente esempio di codice mostra come utilizzare `untag-saml-provider`.

### AWS CLI

Per rimuovere un tag da un SAML provider

Il `untag-saml-provider` comando seguente rimuove qualsiasi tag con il nome chiave 'Department' dal profilo di istanza specificato.

```
aws iam untag-saml-provider \  
  --role-name my-role \  
  --tag-keys Department
```

```
--saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \  
--tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [UntagSamlProvider](#) in AWS CLI Command Reference.

## untag-server-certificate

Il seguente esempio di codice mostra come utilizzare `untag-server-certificate`.

### AWS CLI

Per rimuovere un tag da un certificato del server

Il `untag-server-certificate` comando seguente rimuove qualsiasi tag con il nome chiave 'Department' dal certificato del server specificato.

```
aws iam untag-server-certificate \  
--server-certificate-name ExampleCertificate \  
--tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [UntagServerCertificate](#) in AWS CLI Command Reference.

## untag-user

Il seguente esempio di codice mostra come utilizzare `untag-user`.

### AWS CLI

Per rimuovere un tag da un utente

Il `untag-user` comando seguente rimuove qualsiasi tag con il nome chiave 'Department' dall'utente specificato.

```
aws iam untag-user \  

```

```
--user-name alice \  
--tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare IAM le risorse](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [UntagUser](#) in AWS CLI Command Reference.

## update-access-key

Il seguente esempio di codice mostra come utilizzare `update-access-key`.

### AWS CLI

Per attivare o disattivare una chiave di accesso per un IAM utente

Il `update-access-key` comando seguente disattiva la chiave di accesso specificata (ID della chiave di accesso e chiave di accesso segreta) per l'IAMutente denominato. Bob

```
aws iam update-access-key \  
  --access-key-id AKIAIOSFODNN7EXAMPLE \  
  --status Inactive \  
  --user-name Bob
```

Questo comando non produce alcun output.

La disattivazione della chiave significa che non può essere utilizzata per l'accesso programmatico a. AWS La chiave, tuttavia, rimane disponibile e può essere riattivata.

Per ulteriori informazioni, vedere [Gestione delle chiavi di accesso per IAM gli utenti nella Guida per l'AWS IAMutente](#).

- Per API i dettagli, vedere [UpdateAccessKey](#) in AWS CLI Command Reference.

## update-account-password-policy

Il seguente esempio di codice mostra come utilizzare `update-account-password-policy`.

### AWS CLI

Per impostare o modificare la politica corrente in materia di password dell'account

Il `update-account-password-policy` comando seguente imposta la politica delle password in modo che richieda una lunghezza minima di otto caratteri e richieda uno o più numeri nella password.

```
aws iam update-account-password-policy \  
  --minimum-password-length 8 \  
  --require-numbers
```

Questo comando non produce alcun output.

Le modifiche alla politica in materia di password di un account influiscono su tutte le nuove password create per IAM gli utenti dell'account. Le modifiche alle politiche relative alle password non influiscono sulle password esistenti.

Per ulteriori informazioni, vedere [Impostazione di una politica di password dell'account per IAM gli utenti](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [UpdateAccountPasswordPolicy](#) in AWS CLI Command Reference.

## update-assume-role-policy

Il seguente esempio di codice mostra come utilizzare `update-assume-role-policy`.

### AWS CLI

Per aggiornare la politica di fiducia per un IAM ruolo

Il `update-assume-role-policy` comando seguente aggiorna la politica di fiducia per il ruolo denominato `Test-Role`.

```
aws iam update-assume-role-policy \  
  --role-name Test-Role \  
  --policy-document file:///Test-Role-Trust-Policy.json
```

Questo comando non produce alcun output.

La politica di fiducia è definita come JSON documento nel file `Test-Role-trust-policy.json`. (Il nome e l'estensione del file non hanno importanza.) La policy di attendibilità deve specificare un principale.

Per aggiornare la politica delle autorizzazioni per un ruolo, usa il comando `put-role-policy`



Per ulteriori informazioni, vedere [Creazione di IAM ruoli](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [UpdateAssumeRolePolicy](#) in AWS CLI Command Reference.

## update-group

Il seguente esempio di codice mostra come utilizzare `update-group`.

### AWS CLI

Per rinominare un gruppo IAM

Il `update-group` comando seguente modifica il nome del IAM gruppo `Test` in `Test-1`.

```
aws iam update-group \  
  --group-name Test \  
  --new-group-name Test-1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Ridenominazione di un gruppo di IAM utenti](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [UpdateGroup](#) in AWS CLI Command Reference.

## update-login-profile

Il seguente esempio di codice mostra come utilizzare `update-login-profile`.

### AWS CLI

Per aggiornare la password di un IAM utente

Il `update-login-profile` comando seguente crea una nuova password per l'IAMutente denominato `Bob`.

```
aws iam update-login-profile \  
  --user-name Bob \  
  --password <password>
```

Questo comando non produce alcun output.

Per impostare una politica di password per l'account, utilizzare il `update-account-password-policy` comando. Se la nuova password viola la politica relativa alle password dell'account, il comando restituisce un `PasswordPolicyViolation` errore.

Se la politica sulla password dell'account lo consente, IAM gli utenti possono modificare le proprie password utilizzando il `change-password` comando.

Conserva la password in un luogo sicuro. Se la password viene persa, non può essere recuperata ed è necessario crearne una nuova utilizzando il `create-login-profile` comando.

Per ulteriori informazioni, vedere [Gestione delle password per IAM gli utenti](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [UpdateLoginProfile](#) in AWS CLI Command Reference.

## **update-open-id-connect-provider-thumbprint**

Il seguente esempio di codice mostra come utilizzare `update-open-id-connect-provider-thumbprint`.

### AWS CLI

Per sostituire l'elenco esistente di impronte digitali dei certificati del server con un nuovo elenco

Questo esempio aggiorna l'elenco delle impronte digitali dei certificati per il OIDC provider che `arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com` deve utilizzare ARN una nuova impronta personale.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creazione di provider di identità OpenID Connect \(OIDC\) nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, consultate [UpdateOpenIdConnectProviderThumbprint AWS CLI Command Reference](#).

## update-role-description

Il seguente esempio di codice mostra come utilizzare `update-role-description`.

### AWS CLI

Per modificare la descrizione di un IAM ruolo

Il `update-role` comando seguente modifica la descrizione del IAM ruolo `production-role` in `Main production role`.

```
aws iam update-role-description \  
  --role-name production-role \  
  --description 'Main production role'
```

Output:

```
{  
  "Role": {  
    "Path": "/",  
    "RoleName": "production-role",  
    "RoleId": "ARO1234567890EXAMPLE",  
    "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Modifica di un ruolo](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [UpdateRoleDescription](#) in AWS CLI Command Reference.

## update-role

Il seguente esempio di codice mostra come utilizzare `update-role`.

### AWS CLI

Per modificare la descrizione di un IAM ruolo o la durata della sessione

Il `update-role` comando seguente modifica la descrizione del IAM ruolo `production-role` in `Main production role` e imposta la durata massima della sessione su 12 ore.

```
aws iam update-role \  
  --role-name production-role \  
  --description 'Main production role' \  
  --max-session-duration 43200
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Modifica di un ruolo](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [UpdateRole](#) in AWS CLI Command Reference.

## update-saml-provider

Il seguente esempio di codice mostra come utilizzare `update-saml-provider`.

### AWS CLI

Per aggiornare il documento di metadati per un provider esistente SAML

Questo esempio aggiorna il SAML provider in IAM cui ARN si trova `arn:aws:iam::123456789012:saml-provider/SAMLADFS` con un nuovo documento di SAML metadati tratto dal file `SAMLMetaData.xml`

```
aws iam update-saml-provider \  
  --saml-metadata-document file://SAMLMetaData.xml \  
  --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

Output:

```
{
  "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/SAMLADFS"
}
```

Per ulteriori informazioni, vedete [Creazione di provider di IAM SAML identità](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [UpdateSamlProvider](#) in AWS CLI Command Reference.

## update-server-certificate

Il seguente esempio di codice mostra come utilizzare `update-server-certificate`.

### AWS CLI

Per modificare il percorso o il nome di un certificato del server nel tuo AWS account

Il comando `update-server-certificate` seguente modifica il nome del certificato da `myServerCertificate` a `myUpdatedServerCertificate`. Cambia anche il percorso in `/cloudfront/` modo che sia possibile accedervi dal CloudFront servizio Amazon. Questo comando non produce alcun output. Puoi visualizzare i risultati dell'aggiornamento eseguendo il comando `list-server-certificates`.

```
aws-iam update-server-certificate \
  --server-certificate-name myServerCertificate \
  --new-server-certificate-name myUpdatedServerCertificate \
  --new-path /cloudfront/
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [la sezione Gestione dei certificati del server IAM nella Guida AWS IAM per l'utente](#).

- Per API i dettagli, vedere [UpdateServerCertificate](#) in AWS CLI Command Reference.

## update-service-specific-credential

Il seguente esempio di codice mostra come utilizzare `update-service-specific-credential`.

### AWS CLI

Esempio 1: per aggiornare lo stato della credenziale specifica del servizio dell'utente richiedente

L'update-service-specific-credential esempio seguente modifica lo stato della credenziale specificata per l'utente a cui effettua la richiesta. *Inactive*

```
aws iam update-service-specific-credential \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \  
  --status Inactive
```

Questo comando non produce alcun output.

Esempio 2: Aggiornare lo stato delle credenziali specifiche del servizio di un utente specificato

L'update-service-specific-credential esempio seguente modifica lo stato della credenziale dell'utente specificato in *Inattivo*.

```
aws iam update-service-specific-credential \  
  --user-name sofia \  
  --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \  
  --status Inactive
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creare credenziali Git per le HTTPS connessioni a CodeCommit](#) nella Guida per l'AWS CodeCommit utente

- Per API i dettagli, vedere [UpdateServiceSpecificCredential](#) in AWS CLI Command Reference.

## update-signing-certificate

Il seguente esempio di codice mostra come utilizzare `update-signing-certificate`.

### AWS CLI

Per attivare o disattivare un certificato di firma per un utente IAM

Il `update-signing-certificate` comando seguente disattiva il certificato di firma specificato per l'IAM utente denominato. *Bob*

```
aws iam update-signing-certificate \  
  --certificate-id TA7SMP42TDN5Z260BPJE7EXAMPLE \  
  --status Inactive \  
  --user-name Bob
```

Per ottenere l'ID per un certificato di firma, usa il `list-signing-certificates` comando.

Per ulteriori informazioni, consulta [Gestire i certificati di firma](#) nella Amazon EC2 User Guide.

- Per API i dettagli, consulta [UpdateSigningCertificate AWS CLI Command Reference](#).

## update-ssh-public-key

Il seguente esempio di codice mostra come utilizzare `update-ssh-public-key`.

### AWS CLI

Per modificare lo stato di una chiave SSH pubblica

Il `update-ssh-public-key` comando seguente modifica lo stato della chiave pubblica specificata in `Inactive`.

```
aws iam update-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-id APKA1234567890EXAMPLE \  
  --status Inactive
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Use SSH keys and SSH with CodeCommit](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [UpdateSshPublicKey](#) in AWS CLI Command Reference.

## update-user

Il seguente esempio di codice mostra come utilizzare `update-user`.

### AWS CLI

Per modificare il nome di un IAM utente

Il `update-user` comando seguente modifica il nome dell'IAM utente Bob in Robert.

```
aws iam update-user \  
  --user-name Bob \  
  --user-name Robert
```

```
--new-user-name Robert
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Ridenominazione di un gruppo di IAM utenti](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [UpdateUser](#) in AWS CLI Command Reference.

## upload-server-certificate

Il seguente esempio di codice mostra come utilizzare `upload-server-certificate`.

### AWS CLI

Per caricare un certificato del server sul tuo AWS account

Il `upload-server-certificate` comando seguente carica un certificato server sul tuo AWS account. In questo esempio, il certificato è nel file `public_key_cert_file.pem`, la chiave privata associata è nel file `my_private_key.pem` e la catena di certificati fornita dall'autorità di certificazione (CA) è nel file `my_certificate_chain_file.pem`. Al termine del caricamento, il file è disponibile sotto il nome `myServerCertificate`. I parametri che iniziano con `file://` indicano al comando di leggere il contenuto del file e di utilizzarlo come valore del parametro in luogo del nome del file.

```
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
```

Output:

```
{  
  "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"  }}
```



```
}  
}
```

Per ulteriori informazioni, consulta [Creazione, caricamento ed eliminazione dei certificati server](#) nella guida all'uso. IAM

- Per API i dettagli, vedere [UploadServerCertificate](#) in AWS CLI Command Reference.

## upload-signing-certificate

Il seguente esempio di codice mostra come utilizzare `upload-signing-certificate`.

### AWS CLI

Per caricare un certificato di firma per un IAM utente

Il `upload-signing-certificate` comando seguente carica un certificato di firma per l'IAM utente denominato `Bob`.

```
aws iam upload-signing-certificate \  
  --user-name Bob \  
  --certificate-body file://certificate.pem
```

Output:

```
{  
  "Certificate": {  
    "UserName": "Bob",  
    "Status": "Active",  
    "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----END  
CERTIFICATE-----",  
    "CertificateId": "TA7SMP42TDN5Z260BPJE7EXAMPLE",  
    "UploadDate": "2013-06-06T21:40:08.121Z"  
  }  
}
```

Il certificato si trova in un file denominato `certificate.pem` in formato. PEM

Per ulteriori informazioni, consulta [Creazione e caricamento di un certificato di firma utente](#) nella guida all'uso. IAM

- Per API i dettagli, vedere [UploadSigningCertificate](#) in AWS CLI Command Reference.

## upload-ssh-public-key

Il seguente esempio di codice mostra come utilizzare `upload-ssh-public-key`.

### AWS CLI

Per caricare una chiave SSH pubblica e associarla a un utente

Il `upload-ssh-public-key` comando seguente carica la chiave pubblica trovata nel file `sshkey.pub` e la allega all'utente `sofia`

```
aws iam upload-ssh-public-key \  
  --user-name sofia \  
  --ssh-public-key-body file://sshkey.pub
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni su come generare chiavi in un formato adatto a questo comando, vedere [SSHe Linux, macOS o Unix: Configurare le chiavi pubbliche e private per Git CodeCommit e/o e Windows: configurare le chiavi pubbliche SSH e private per Git e CodeCommit nella Guida per l'AWS CodeCommit utente](#).

- Per API i dettagli, consulta AWS CLI Command [UploadSshPublicKeyReference](#).

## IAM Esempi di Access Analyzer utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with IAM Access Analyzer.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni illustrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto degli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **apply-archive-rule**

Il seguente esempio di codice mostra come utilizzare `apply-archive-rule`.

AWS CLI

Per applicare una regola di archiviazione ai risultati esistenti che soddisfano i criteri della regola di archiviazione

L'`apply-archive-rule` esempio seguente applica una regola di archiviazione ai risultati esistenti che soddisfano i criteri della regola di archiviazione.

```
aws accessanalyzer apply-archive-rule \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyArchiveRule
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [le regole di archiviazione](#) nella Guida AWS IAM per l'utente.

- Per API i dettagli, vedere [ApplyArchiveRule](#) in AWS CLI Command Reference.

### **cancel-policy-generation**

Il seguente esempio di codice mostra come utilizzare `cancel-policy-generation`.

AWS CLI

Per annullare la generazione della politica richiesta

L'`cancel-policy-generation` esempio seguente annulla l'ID del lavoro richiesto per la generazione delle politiche.

```
aws accessanalyzer cancel-policy-generation \  
  --job-id 923a56b0-ebb8-4e80-8a3c-a11ccfbcd6f2
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Generazione di policy di IAM Access Analyzer](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [CancelPolicyGeneration](#) in AWS CLI Command Reference.

## check-access-not-granted

Il seguente esempio di codice mostra come utilizzare `check-access-not-granted`.

### AWS CLI

Per verificare se l'accesso specificato non è consentito da una politica

L'`check-access-not-granted` esempio seguente verifica se l'accesso specificato non è consentito da una politica.

```
aws accessanalyzer check-access-not-granted \  
  --policy-document file://myfile.json \  
  --access-actions="s3:DeleteBucket","s3:GetBucketLocation" \  
  --policy-type IDENTITY_POLICY
```

Contenuto di `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/*"
  ]
}

```

Output:

```

{
  "result": "PASS",
  "message": "The policy document does not grant access to perform one or more of
the listed actions."
}

```

Per ulteriori informazioni, vedere [Anteprima dell'accesso con IAM Access Analyzer APIs nella Guida](#) per l'AWS IAM utente.

- Per API i dettagli, vedere [CheckAccessNotGranted](#) in AWS CLI Command Reference.

## check-no-new-access

Il seguente esempio di codice mostra come utilizzare `check-no-new-access`.

### AWS CLI

Per verificare se è consentito un nuovo accesso per una politica aggiornata rispetto alla politica esistente

L'`check-no-new-access` seguente verifica se è consentito un nuovo accesso per una politica aggiornata rispetto alla politica esistente.

```

aws accessanalyzer check-no-new-access \
  --existing-policy-document file://existing-policy.json \
  --new-policy-document file://new-policy.json \
  --policy-type IDENTITY_POLICY

```

Contenuto di `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/*"
  ]
}
```

Contenuto di 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/*"
      ]
    }
  ]
}
```

Output:

```
{
  "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
      }
    ]
  }

```

Per ulteriori informazioni, vedere [Anteprima dell'accesso con IAM Access Analyzer APIs nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [CheckNoNewAccess](#) in AWS CLI Command Reference.

## check-no-public-access

Il seguente esempio di codice mostra come utilizzare `check-no-public-access`.

### AWS CLI

Per verificare se una politica delle risorse può concedere l'accesso pubblico al tipo di risorsa specificato

L'`check-no-public-access`esempio seguente verifica se una politica delle risorse può concedere l'accesso pubblico al tipo di risorsa specificato.

```

aws accessanalyzer check-no-public-access \
  --policy-document file://check-no-public-access-myfile.json \
  --resource-type AWS::S3::Bucket

```

Contenuto di `myfile.json`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CheckNoPublicAccess",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::111122223333:user/JohnDoe" },
      "Action": [
        "s3:GetObject"
      ]
    }
  ]
}

```

**Output:**

```
{
  "result": "PASS",
  "message": "The resource policy does not grant public access for the given
resource type."
}
```

Per ulteriori informazioni, vedere [Anteprima dell'accesso con IAM Access Analyzer APIs nella Guida](#) per l'AWS IAM utente.

- Per API i dettagli, vedere [CheckNoPublicAccess](#) in AWS CLI Command Reference.

**create-access-preview**

Il seguente esempio di codice mostra come utilizzare `create-access-preview`.

**AWS CLI**

Per creare un'anteprima di accesso che consenta di visualizzare in anteprima i risultati di IAM Access Analyzer relativi alla risorsa prima di distribuire le autorizzazioni delle risorse

L'esempio seguente crea un'anteprima di accesso che consente di visualizzare in anteprima i risultati di IAM Access Analyzer relativi alla risorsa prima di distribuire le autorizzazioni delle risorse nell'account. AWS

```
aws accessanalyzer create-access-preview \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \
  --configurations file://myfile.json
```

Contenuto di `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"
        }
    ]
}
}
}
}
}

```

Output:

```

{
  "id": "3c65eb13-6ef9-4629-8919-a32043619e6b"
}

```

Per ulteriori informazioni, vedere [Anteprima dell'accesso con IAM Access Analyzer APIs](#) nella Guida per l'utente.AWS IAM

- Per API i dettagli, vedere [CreateAccessPreview](#) in AWS CLI Command Reference.

## create-analyzer

Il seguente esempio di codice mostra come utilizzare `create-analyzer`.

AWS CLI

Per creare un analizzatore

L'`create-analyzer` esempio seguente crea un analizzatore nel tuo AWS account.

```

aws accessanalyzer create-analyzer \
  --analyzer-name example \
  --type ACCOUNT

```

Output:

```
{
  "arn": "arn:aws:access-analyzer:us-east-2:111122223333:analyzer/example"
}
```

Per ulteriori informazioni, consulta la [Guida introduttiva ai risultati di AWS Identity and Access Management Access Analyzer](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [CreateAnalyzer](#) in AWS CLI Command Reference.

## create-archive-rule

Il seguente esempio di codice mostra come utilizzare `create-archive-rule`.

### AWS CLI

Per creare una regola di archiviazione per l'analizzatore specificato

L'`create-archive-rule` esempio seguente crea una regola di archiviazione per l'analizzatore specificato nel tuo AWS account.

```
aws accessanalyzer create-archive-rule \
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
  --rule-name MyRule \
  --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq": ["AWS::IAM::Role"]}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [le regole di archiviazione](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [CreateArchiveRule](#) in AWS CLI Command Reference.

## delete-analyzer

Il seguente esempio di codice mostra come utilizzare `delete-analyzer`.

### AWS CLI

Per eliminare l'analizzatore specificato

L'`delete-analyzer` esempio seguente elimina l'analizzatore specificato nell'account. AWS

```
aws accessanalyzer delete-analyzer \  
  --analyzer-name example
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta le [regole di archiviazione nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [DeleteAnalyzer](#) in AWS CLI Command Reference.

## delete-archive-rule

Il seguente esempio di codice mostra come utilizzare `delete-archive-rule`.

AWS CLI

Per eliminare la regola di archiviazione specificata

L'`delete-archive-rule` esempio seguente elimina la regola di archiviazione specificata nell'AWS account.

```
aws accessanalyzer delete-archive-rule \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyRule
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [le regole di archiviazione](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [DeleteArchiveRule](#) in AWS CLI Command Reference.

## get-access-preview

Il seguente esempio di codice mostra come utilizzare `get-access-preview`.

AWS CLI

Per recuperare informazioni su un'anteprima di accesso per l'analizzatore specificato

L'`get-access-preview` esempio seguente recupera le informazioni su un'anteprima di accesso per l'analizzatore specificato nel tuo account. AWS

```
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
```

Output:

```
{
  "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": "{\"Version\":\"2012-10-17\",\"Statement\":\n\n[\n\n{\n\n\"Effect\":\"Allow\", \"Principal\":{\"AWS\":[\n\n\"arn:aws:iam::111122223333:root\"]},\n\n\"Action\":[\n\n\"s3:PutObject\", \"s3:PutObjectAcl\"], \"Resource\": \"arn:aws:s3::DOC-EXAMPLE-BUCKET/*\"}]\n\n}\"",
          "bucketAclGrants": [
            {
              "permission": "READ",
              "grantee": {
                "id":
"79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"
            }
          ]
        },
        "bucketPublicAccessBlock": {
          "ignorePublicAcls": true,
          "restrictPublicBuckets": true
        }
      }
    }
  },
  "createdAt": "2024-02-17T00:18:44+00:00",
  "status": "COMPLETED"
}
```

Per ulteriori informazioni, vedere [Anteprima dell'accesso con IAM Access Analyzer APIs](#) nella Guida per l'utente.AWS IAM

- Per API i dettagli, vedere [GetAccessPreview](#) in AWS CLI Command Reference.

## get-analyzed-resource

Il seguente esempio di codice mostra come utilizzare `get-analyzed-resource`.

### AWS CLI

Per recuperare informazioni su una risorsa che è stata analizzata

L'`get-analyzed-resource` esempio seguente recupera informazioni su una risorsa che è stata analizzata nel tuo AWS account.

```
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
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, vedere [Using AWS Identity and Access Management Access Analyzer](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [GetAnalyzedResource](#) in AWS CLI Command Reference.

## get-analyzer

Il seguente esempio di codice mostra come utilizzare `get-analyzer`.

### AWS CLI

Per recuperare informazioni sull'analizzatore specificato

L'`get-analyzer` esempio seguente recupera le informazioni sull'analizzatore specificato nel tuo account. AWS

```
aws accessanalyzer get-analyzer \  
  --analyzer-name ConsoleAnalyzer-account
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Using AWS Identity and Access Management Access Analyzer](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [GetAnalyzer](#) in AWS CLI Command Reference.

## get-archive-rule

Il seguente esempio di codice mostra come utilizzare `get-archive-rule`.

AWS CLI

Per recuperare informazioni su una regola di archiviazione

L'`get-archive-rule` esempio seguente recupera informazioni su una regola di archiviazione nel tuo AWS account.

```
aws accessanalyzer get-archive-rule \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-id
```

```
--rule-name MyArchiveRule
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [le regole di archiviazione](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [GetArchiveRule](#) in AWS CLI Command Reference.

## get-finding-v2

Il seguente esempio di codice mostra come utilizzare `get-finding-v2`.

### AWS CLI

Per recuperare informazioni sul risultato specificato

L'`get-finding-v2` esempio seguente recupera informazioni sul risultato specificato nel tuo account. AWS

```
aws accessanalyzer get-finding-v2 \
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-organization \
```

```
--id 0910eedb-381e-4e95-adda-0d25c19e6e90
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta la sezione [Revisione dei risultati](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [GetFindingV2](#) in AWS CLI Command Reference.

## get-finding

Il seguente esempio di codice mostra come utilizzare `get-finding`.



## AWS CLI

Per recuperare informazioni sul risultato specificato

L'get-findingesempio seguente recupera informazioni sul risultato specificato nel tuo account.  
AWS

```
aws accessanalyzer get-finding \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-organization \  
  --id 0910eedb-381e-4e95-adda-0d25c19e6e90
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta la sezione [Revisione dei risultati](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [GetFinding](#) in AWS CLI Command Reference.

## get-generated-policy

Il seguente esempio di codice mostra come utilizzare `get-generated-policy`.

### AWS CLI

Per recuperare la politica che è stata generata utilizzando il comando `StartPolicyGeneration` API

L'esempio seguente recupera la policy che è stata generata utilizzando il `StartPolicyGeneration` API nel tuo AWS account.

```
aws accessanalyzer get-generated-policy \
  --job-id c557dc4a-0338-4489-95dd-739014860ff9
```

Output:

```
{
  "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": []
    }
]
},
"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"
}
}
}

```

Per ulteriori informazioni, vedere [Generazione di policy di IAM Access Analyzer](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [GetGeneratedPolicy](#) in AWS CLI Command Reference.

## list-access-preview-findings

Il seguente esempio di codice mostra come utilizzare `list-access-preview-findings`.

### AWS CLI

Per recuperare un elenco dei risultati dell'anteprima di accesso generati dall'anteprima di accesso specificata

L'`list-access-preview-findings` esempio seguente recupera un elenco di risultati di anteprima di accesso generati dall'anteprima di accesso specificata nel tuo AWS account.

```

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

```

Output:

```
{
```

```

    "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"
          }
        ]
      }
    ]
  }
}

```

Per ulteriori informazioni, vedere [Visualizzazione dell'anteprima dell'accesso con IAM Access Analyzer APIs](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [ListAccessPreviewFindings](#) in AWS CLI Command Reference.

## list-access-previews

Il seguente esempio di codice mostra come utilizzare `list-access-previews`.

### AWS CLI

Per recuperare un elenco di anteprime di accesso per l'analizzatore specificato

L'`list-access-previews`esempio seguente recupera un elenco di anteprime di accesso per l'analizzatore specificato nel tuo account. AWS

```
aws accessanalyzer list-access-previews \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Visualizzazione dell'anteprima dell'accesso con IAM Access Analyzer APIs](#) nella Guida per l'utente.AWS IAM

- Per API i dettagli, vedere [ListAccessPreviews](#) in AWS CLI Command Reference.

## list-analyzed-resources

Il seguente esempio di codice mostra come utilizzare `list-analyzed-resources`.

AWS CLI

Per elencare i widget disponibili

L'`list-analyzed-resources` esempio seguente elenca i widget disponibili nel tuo AWS account.

```
aws accessanalyzer list-analyzed-resources \  
  --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \  
  --resource-type AWS::IAM::Role
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Using AWS Identity and Access Management Access Analyzer](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [ListAnalyzedResources](#) in AWS CLI Command Reference.

## list-analyzers

Il seguente esempio di codice mostra come utilizzare `list-analyzers`.

### AWS CLI

Per recuperare un elenco di analizzatori

L'`list-analyzers` esempio seguente recupera un elenco di analizzatori presenti nel tuo account. AWS

```
aws accessanalyzer list-analyzers
```

## Output:

```
{
  "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"
    }
  ]
}
```

```
}
```

Per ulteriori informazioni, vedere [Using AWS Identity and Access Management Access Analyzer](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [ListAnalyzers](#) in AWS CLI Command Reference.

## list-archive-rules

Il seguente esempio di codice mostra come utilizzare `list-archive-rules`.

### AWS CLI

Per recuperare un elenco di regole di archiviazione create per l'analizzatore specificato

L'`list-archive-rules` esempio seguente recupera un elenco di regole di archiviazione create per l'analizzatore specificato nell'account. AWS

```
aws accessanalyzer list-archive-rules \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization
```

Output:

```
{  
  "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"
    }
  ]
}

```

Per ulteriori informazioni, vedere [Using AWS Identity and Access Management Access Analyzer](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [ListArchiveRules](#) in AWS CLI Command Reference.

## list-findings-v2

Il seguente esempio di codice mostra come utilizzare `list-findings-v2`.

### AWS CLI

Per recuperare un elenco di risultati generati dall'analizzatore specificato

L'`list-findings-v2` esempio seguente recupera un elenco di risultati generati dall'analizzatore specificato nel tuo account. AWS Questo esempio filtra i risultati per includere solo i IAM ruoli il cui nome contiene. 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"]}}'

```

Output:

```
{
```

```
"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"
  }
]
```

Per ulteriori informazioni, vedere [Using AWS Identity and Access Management Access Analyzer](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [ListFindingsV2](#) in AWS CLI Command Reference.

## list-findings

Il seguente esempio di codice mostra come utilizzare `list-findings`.

### AWS CLI

Per recuperare un elenco di risultati generati dall'analizzatore specificato

L'esempio seguente recupera un elenco di risultati generati dall'analizzatore specificato nel tuo account. AWS Questo esempio filtra i risultati per includere solo i IAM ruoli il cui nome contiene 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"]}]}'
```

Output:

```
{  
  "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"
    }
]
}
```

Per ulteriori informazioni, vedere [Using AWS Identity and Access Management Access Analyzer](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [ListFindings](#) in AWS CLI Command Reference.

## list-policy-generations

Il seguente esempio di codice mostra come utilizzare `list-policy-generations`.

### AWS CLI

Per elencare tutte le generazioni di policy richieste negli ultimi sette giorni

L'`list-policy-generations` esempio seguente elenca tutte le generazioni di policy richieste nel tuo AWS account negli ultimi sette giorni.

```
aws accessanalyzer list-policy-generations
```

Output:

```
{
  "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"
    }
]
}
```

Per ulteriori informazioni, consulta la sezione [Generazione di policy di IAM Access Analyzer](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [ListPolicyGenerations](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per recuperare un elenco di tag applicati alla risorsa specificata

L'`list-tags-for-resource`esempio seguente recupera un elenco di tag applicati alla risorsa specificata nel tuo AWS account.

```
aws accessanalyzer list-tags-for-resource \
  --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account
```

Output:

```
{
  "tags": {
    "Zone-of-trust": "Account",
    "Name": "ConsoleAnalyzer"
  }
}
```

Per ulteriori informazioni, vedere [Generazione di policy di IAM Access Analyzer](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## start-policy-generation

Il seguente esempio di codice mostra come utilizzare `start-policy-generation`.

### AWS CLI

Per avviare una richiesta di generazione di policy

L'esempio seguente avvia una richiesta di generazione di policy nel tuo AWS account.

```
aws accessanalyzer start-policy-generation \  
  --policy-generation-details '{"principalArn":"arn:aws:iam::111122223333:role/  
Admin"}' \  
  --cloud-trail-details file://myfile.json
```

Contenuto di `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"  
    }  
  ]  
}
```

Output:

```
{  
  "jobId": "c557dc4a-0338-4489-95dd-739014860ff9"  
}
```

Per ulteriori informazioni, vedere [Generazione di policy di IAM Access Analyzer](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [StartPolicyGeneration](#) in AWS CLI Command Reference.

## start-resource-scan

Il seguente esempio di codice mostra come utilizzare `start-resource-scan`.

### AWS CLI

Per avviare immediatamente una scansione delle politiche applicate alla risorsa specificata

L'`start-resource-scan`esempio seguente avvia immediatamente una scansione delle politiche applicate alla risorsa specificata nel tuo AWS account.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Generazione di policy di IAM Access Analyzer nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [StartResourceScan](#)in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere un tag alla risorsa specificata

L'`tag-resource`esempio seguente aggiunge un tag alla risorsa specificata nell' AWS account.

```
aws accessanalyzer tag-resource \  
  --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/  
ConsoleAnalyzer-account \  
  --tags Environment=dev,Purpose=testing
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Using AWS Identity and Access Management Access Analyzer](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [TagResource](#)in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag dalle risorse specificate

L'`untag-resource` esempio seguente rimuove i tag dalla risorsa specificata nel tuo AWS account.

```
aws accessanalyzer untag-resource \  
  --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/ConsoleAnalyzer-account \  
  --tag-keys Environment Purpose
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Using AWS Identity and Access Management Access Analyzer](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-archive-rule

Il seguente esempio di codice mostra come utilizzare `update-archive-rule`.

### AWS CLI

Per aggiornare i criteri e i valori per la regola di archiviazione specificata

L'`update-archive-rule` esempio seguente aggiorna i criteri e i valori per la regola di archiviazione specificata nel tuo AWS account.

```
aws accessanalyzer update-archive-rule \  
  --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \  
  --rule-name MyArchiveRule \  
  --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq": ["AWS::IAM::Role"]}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [le regole di archiviazione](#) nella Guida AWS IAM per l'utente.



- Per API i dettagli, vedere [UpdateArchiveRule](#) in AWS CLI Command Reference.

## update-findings

Il seguente esempio di codice mostra come utilizzare `update-findings`.

### AWS CLI

Per aggiornare lo stato dei risultati specificati

L'`update-findings` esempio seguente aggiorna lo stato dei risultati specificati nel tuo AWS account.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Using AWS Identity and Access Management Access Analyzer](#) nella Guida per l'AWS IAM utente.

- Per API i dettagli, vedere [UpdateFindings](#) in AWS CLI Command Reference.

## validate-policy

Il seguente esempio di codice mostra come utilizzare `validate-policy`.

### AWS CLI

Per richiedere la convalida di una politica e restituire un elenco di risultati

L'`validate-policy` esempio seguente richiede la convalida di una politica e restituisce un elenco di risultati. La policy dell'esempio è una policy di trust dei ruoli per un ruolo di Amazon Cognito utilizzato per la federazione delle identità web. I risultati generati dalla politica di fiducia si riferiscono a un valore di `Sid` elemento vuoto e a un principio di policy non corrispondente a causa dell'uso errato dell'azione di assunzione del ruolo, `sts:AssumeRole`. L'azione di assunzione del ruolo corretta da utilizzare con Cognito è `sts:AssumeRoleWithWebIdentity`.

```
aws accessanalyzer validate-policy \  
  --policy-document file://myfile.json \  
  --policy-type RESOURCE_POLICY
```

Contenuto di `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"  
        }  
      }  
    }  
  ]  
}
```

Output:

```
{  
  "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": "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
      }
    }
  }
]
}
```

Per ulteriori informazioni, consulta [Controlli per la convalida delle politiche nella Guida](#) per l'AWS IAMutente.

- Per API i dettagli, vedere [ValidatePolicy](#) in AWS CLI Command Reference.

## Esempi di Image Builder che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Image Builder.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **create-component**

Il seguente esempio di codice mostra come utilizzare `create-component`.

#### AWS CLI

Per creare un componente

L'esempio seguente crea un componente che utilizza un file di JSON documento e fa riferimento a un documento componente in un YAML formato che viene caricato in un bucket Amazon S3.

```
aws imagebuilder create-component \  
  --cli-input-json file://create-component.json
```

Contenuto di `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"
}
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [CreateComponent](#) in AWS CLI Command Reference.

## create-distribution-configuration

Il seguente esempio di codice mostra come utilizzare `create-distribution-configuration`.

AWS CLI

Per creare una configurazione di distribuzione

L'`create-distribution-configuration` esempio seguente crea una configurazione di distribuzione utilizzando un JSON file.

```
aws imagebuilder create-distribution-configuration \
  --cli-input-json file:/create-distribution-configuration.json
```

Contenuto di `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"
                ]
            }
        }
    }
]
}

```

### Output:

```

{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution"
}

```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLl](#)utente di EC2Image Builder.

- Per API i dettagli, vedere [CreateDistributionConfiguration](#)in AWS CLI Command Reference.

## create-image-pipeline

Il seguente esempio di codice mostra come utilizzare `create-image-pipeline`.

### AWS CLI

Per creare una pipeline di immagini

L'esempio seguente crea una pipeline di immagini utilizzando un JSON file.

```
aws imagebuilder create-image-pipeline \  
  --cli-input-json file://create-image-pipeline.json
```

Contenuto di `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"  
}
```

Output:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
```

```
"imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline"
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l'utente di EC2Image Builder](#).

- Per API i dettagli, vedere [CreateImagePipeline](#) in AWS CLI Command Reference.

## create-image-recipe

Il seguente esempio di codice mostra come utilizzare `create-image-recipe`.

### AWS CLI

Per creare una ricetta

L'`create-image-recipe` esempio seguente crea una ricetta di immagini utilizzando un JSON file. I componenti vengono installati nell'ordine in cui sono stati specificati.

```
aws imagebuilder create-image-recipe \
  --cli-input-json file://create-image-recipe.json
```

Contenuto di `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"
```

```
}
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#)utente di EC2Image Builder.

- Per API i dettagli, vedere [CreateImageRecipe](#)in AWS CLI Command Reference.

## create-image

Il seguente esempio di codice mostra come utilizzare `create-image`.

AWS CLI

Per creare un'immagine

L'`create-image`esempio seguente crea un'immagine.

```
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
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [CreateImage](#) in AWS CLI Command Reference.

## create-infrastructure-configuration

Il seguente esempio di codice mostra come utilizzare `create-infrastructure-configuration`.

### AWS CLI

Per creare una configurazione dell'infrastruttura

L'`create-infrastructure-configuration` esempio seguente crea una configurazione dell'infrastruttura utilizzando un JSON file.

```
aws imagebuilder create-infrastructure-configuration \  
  --cli-input-json file://create-infrastructure-configuration.json
```

Contenuto di `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"  
}
```

## Output:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [CreateInfrastructureConfiguration](#) in AWS CLI Command Reference.

## delete-component

Il seguente esempio di codice mostra come utilizzare `delete-component`.

### AWS CLI

Per eliminare un componente

L'`delete-component` esempio seguente elimina la versione di build di un componente specificandone la. ARN

```
aws imagebuilder delete-component \
  --component-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:component/myexamplecomponent/2019.12.02/1
```

## Output:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "componentBuildVersionArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/myexamplecomponent/2019.12.02/1"
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [DeleteComponent](#) in AWS CLI Command Reference.

## delete-image-pipeline

Il seguente esempio di codice mostra come utilizzare `delete-image-pipeline`.

### AWS CLI

Per eliminare una pipeline di immagini

L'`delete-image-pipeline`esempio seguente elimina una pipeline di immagini specificandone l'ARN

```
aws imagebuilder delete-image-pipeline \  
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
my-example-pipeline
```

Output:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline"  
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2 Image Builder.

- Per API i dettagli, vedere [DeleteImagePipeline](#) in AWS CLI Command Reference.

## delete-image-recipe

Il seguente esempio di codice mostra come utilizzare `delete-image-recipe`.

### AWS CLI

Per eliminare un'immagine, una ricetta

L'`delete-image-recipe`esempio seguente elimina una ricetta di immagini specificandone l'ARN

```
aws imagebuilder delete-image-recipe \  
  --image-recipe-arn arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/  
mybasicrecipe/2019.12.03
```

**Output:**

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03"
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [DeleteImageRecipe](#) in AWS CLI Command Reference.

**delete-image**

Il seguente esempio di codice mostra come utilizzare `delete-image`.

**AWS CLI**

Per eliminare un'immagine

L'`delete-image` esempio seguente elimina una versione di build dell'immagine specificandone la. ARN

```
aws imagebuilder delete-image \
  --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-
example-image/2019.12.02/1
```

**Output:**

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1"
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [DeleteImage](#) in AWS CLI Command Reference.



## delete-infrastructure-configuration

Il seguente esempio di codice mostra come utilizzare `delete-infrastructure-configuration`.

### AWS CLI

Per eliminare una configurazione dell'infrastruttura

L'esempio seguente elimina una pipeline di immagini specificandone l'ARN.

```
aws imagebuilder delete-infrastructure-configuration \
  --infrastructure-configuration-arn arn:aws:imagebuilder:us-
  east-1:123456789012:infrastructure-configuration/myexampleinfrastructure
```

Output:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
  west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l'AWS CLI](#) utente di EC2 Image Builder.

- Per API i dettagli, vedere [DeleteInfrastructureConfiguration](#) in AWS CLI Command Reference.

## get-component-policy

Il seguente esempio di codice mostra come utilizzare `get-component-policy`.

### AWS CLI

Per ottenere dettagli sulla politica dei componenti

L'esempio seguente elenca i dettagli di una politica dei componenti specificandone l'ARN.

```
aws imagebuilder get-component-policy \
  --component-arn arn:aws:imagebuilder:us-west-2:123456789012:component/my-
  example-component/2019.12.03/1
```

**Output:**

```
{
  "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\" ] } ] }"
}
```

Per ulteriori informazioni, vedere Configurazione e gestione di una pipeline di immagini di EC2 Image Builder utilizzando AWS CLI < <https://docs.aws.amazon.com/imagebuilder/latest/userguide/managing-image-builder-cli.html>> nella Guida per l'utente di Image Builder. EC2

- Per i dettagli, vedere in Command Reference. API [GetComponentPolicy](#) AWS CLI

**get-component**

Il seguente esempio di codice mostra come utilizzare `get-component`.

**AWS CLI**

Per ottenere i dettagli dei componenti

L'`get-component` seguente elenca i dettagli di un componente specificandone ARN.

```
aws imagebuilder get-component \
  --component-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:component/component-name/1.0.0/1
```

**Output:**

```
{
  "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",
      "description": "This is hello world testing document.",
      "schemaVersion": "1.0",
      "phases": [
        {
          "name": "test",
          "steps": [
            {
              "name": "HelloWorldStep",
              "action": "ExecuteBash",
              "inputs": {
                "commands": [
                  "echo \"Hello World! Test.\"",
                  "echo \"Hello World! Test.\"",
                  "echo \"Hello World! Test.\""
                ],
                "encrypted": true,
                "dateCreated": "2020-01-27T20:43:30.306Z",
                "tags": {}
              }
            }
          ]
        }
      ]
    }
  }
}

```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [GetComponent](#) in AWS CLI Command Reference.

## get-distribution-configuration

Il seguente esempio di codice mostra come utilizzare `get-distribution-configuration`.

### AWS CLI

Per ottenere i dettagli di una configurazione di distribuzione

L'esempio seguente mostra i dettagli di una configurazione di distribuzione specificandone ARN.

```

aws imagebuilder get-distribution-configuration \
  --distribution-configuration-arn arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/myexampledistribution

```

Output:

```

{
  "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": {}
    }
}

```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#)utente di EC2Image Builder.

- Per API i dettagli, vedere [GetDistributionConfiguration](#)in AWS CLI Command Reference.

## get-image-pipeline

Il seguente esempio di codice mostra come utilizzare `get-image-pipeline`.

## AWS CLI

Per ottenere i dettagli della pipeline di immagini

L'get-image-pipelineesempio seguente elenca i dettagli di una pipeline di immagini specificandone. ARN

```
aws imagebuilder get-image-pipeline \  
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline
```

Output:

```
{  
  "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": {}  
  }  
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [GetImagePipeline](#) in AWS CLI Command Reference.

## get-image-policy

Il seguente esempio di codice mostra come utilizzare `get-image-policy`.

### AWS CLI

Per ottenere dettagli sulle politiche relative alle immagini

L'`get-image-policy` esempio seguente elenca i dettagli di una politica relativa alle immagini specificandone ARN.

```
aws imagebuilder get-image-policy \  
  --image-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-  
image/2019.12.03/1
```

Output:

```
{  
  "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\" ] } ] }"
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [GetImagePolicy](#) in AWS CLI Command Reference.

## get-image-recipe-policy

Il seguente esempio di codice mostra come utilizzare `get-image-recipe-policy`.

### AWS CLI

Per ottenere dettagli sulla politica relativa alle ricette relative alle immagini

L'get-image-recipe-policy esempio seguente elenca i dettagli di una policy relativa alle ricette relative alle immagini specificandone ARN.

```
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
```

Output:

```
{
  "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\" ] } ] }"
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2 Image Builder.

- Per API i dettagli, vedere [GetImageRecipePolicy](#) in AWS CLI Command Reference.

## get-image

Il seguente esempio di codice mostra come utilizzare get-image.

### AWS CLI

Per ottenere i dettagli dell'immagine

L'get-image esempio seguente elenca i dettagli di un'immagine specificandone ARN.

```
aws imagebuilder get-image \
  --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/mybasicrecipe/2019.12.03/1
```

Output:

```
{
  "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": {}
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [GetImage](#) in AWS CLI Command Reference.

## get-infrastructure-configuration

Il seguente esempio di codice mostra come utilizzare `get-infrastructure-configuration`.

### AWS CLI

Per ottenere dettagli sulla configurazione dell'infrastruttura

L'`get-infrastructure-configuration` esempio seguente elenca i dettagli di una configurazione dell'infrastruttura specificandone ARN.

```
aws imagebuilder get-infrastructure-configuration \
```

```
--infrastructure-configuration-arn arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/myexampleinfrastructure
```

Output:

```
{
  "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": {}
  }
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#)utente di EC2Image Builder.

- Per API i dettagli, vedere [GetInfrastructureConfiguration](#)in AWS CLI Command Reference.

## import-component

Il seguente esempio di codice mostra come utilizzare `import-component`.

## AWS CLI

Per importare un componente

L'import-componentesempio seguente importa uno script preesistente utilizzando un JSON file.

```
aws imagebuilder import-component \  
  --cli-input-json file://import-component.json
```

Contenuto di 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"  
}
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#)utente di EC2Image Builder.

- Per API i dettagli, vedere [ImportComponent](#)in AWS CLI Command Reference.

## list-component-build-versions

Il seguente esempio di codice mostra come utilizzarelist-component-build-versions.

## AWS CLI

Per elencare le versioni di build dei componenti

L'`list-component-build-versions` seguente elenca le versioni di build dei componenti con una versione semantica specifica.

```
aws imagebuilder list-component-build-versions --component-  
version-arn arn:aws:imagebuilder:us-west-2:123456789012:component/  
myexamplecomponent/2019.12.02
```

Output:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [ListComponentBuildVersions](#) in AWS CLI Command Reference.

## list-components

Il seguente esempio di codice mostra come utilizzare `list-components`.

## AWS CLI

Per elencare tutte le versioni semantiche dei componenti

L'`list-components` seguente elenca tutte le versioni semantiche dei componenti a cui è possibile accedere. Facoltativamente, puoi filtrare se elencare i componenti di tua proprietà, di Amazon o che sono stati condivisi con te da altri account.

```
aws imagebuilder list-components
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l'AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [ListComponents](#) in AWS CLI Command Reference.

## list-distribution-configurations

Il seguente esempio di codice mostra come utilizzare `list-distribution-configurations`.

## AWS CLI

Per elencare le distribuzioni

L'`list-distribution-configurations` seguente elenca tutte le tue distribuzioni.

```
aws imagebuilder list-distribution-configurations
```

Output:

```
{
  "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"
      }
    }
  ]
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l'AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [ListDistributionConfigurations](#) in AWS CLI Command Reference.

## list-image-build-versions

Il seguente esempio di codice mostra come utilizzare `list-image-build-versions`.

AWS CLI

Per elencare le versioni di build delle immagini

L'`list-image-build-versions` esempio seguente elenca tutte le versioni di build dell'immagine con una versione semantica.

```
aws imagebuilder list-image-build-versions \
  --image-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/  
mybasicrecipe/2019.12.03
```

Output:

```

{
  "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\", \"ResponseCode\": 1, \"Output\": \"\
\n-----ERROR-----\nfailed to run commands: exit status 1\", \"CommandId\":
\"a1b2c3d4-5678-90ab-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": {}
    }
  ]
}

```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [ListImageBuildVersions](#) in AWS CLI Command Reference.

## list-image-pipeline-images

Il seguente esempio di codice mostra come utilizzare `list-image-pipeline-images`.

### AWS CLI

Per elencare le immagini della pipeline: le immagini della pipeline

L'`list-image-pipeline-images` seguente elenca tutte le immagini create da una pipeline di immagini specifica.

```
aws imagebuilder list-image-pipeline-images \
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

Output:

```
{
  "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"
        }
    }
]
}

```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#)utente di EC2Image Builder.

- Per API i dettagli, vedere [ListImagePipelineImages](#)in AWS CLI Command Reference.

## list-image-recipes

Il seguente esempio di codice mostra come utilizzare `list-image-recipes`.

### AWS CLI

Per elencare le ricette con immagini

L'`list-image-recipes`esempio seguente elenca tutte le tue ricette di immagini.

```
aws imagebuilder list-image-recipes
```

Output:

```

{
  "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"
      }
    }
  ]
}

```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [ListImageRecipes](#) in AWS CLI Command Reference.

## list-images

Il seguente esempio di codice mostra come utilizzare `list-images`.

### AWS CLI

Per elencare le immagini

L'`list-images` esempio seguente elenca tutte le versioni semantiche a cui hai accesso.

```
aws imagebuilder list-images
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [ListImages](#) in AWS CLI Command Reference.

## list-infrastructure-configurations

Il seguente esempio di codice mostra come utilizzare `list-infrastructure-configurations`.

AWS CLI

Per elencare le configurazioni dell'infrastruttura

L'`list-infrastructure-configuration` esempio seguente elenca tutte le configurazioni dell'infrastruttura.

```
aws imagebuilder list-infrastructure-configurations
```

Output:

```
{
```

```

    "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"
        }
      }
    ]
  }
}

```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [ListInfrastructureConfigurations](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

AWS CLI

Per elencare i tag per una risorsa specifica

L'`list-tags-for-resource` esempio seguente elenca tutti i tag per una risorsa specifica.

```

aws imagebuilder list-tags-for-resource \
  --resource-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline

```

Output:

```
{
  "tags": {
    "KeyName": "KeyValue"
  }
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l'AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## put-component-policy

Il seguente esempio di codice mostra come utilizzare `put-component-policy`.

### AWS CLI

Per applicare una politica delle risorse a un componente

Il `put-component-policy` comando seguente applica una politica delle risorse a un componente di compilazione per consentire la condivisione tra account dei componenti di compilazione. Ti consigliamo di utilizzare il RAM CLI comando `create-resource-share`. Se si utilizza il CLI comando EC2 Image Builder `put-component-policy`, è necessario utilizzare anche il RAM CLI comando `promote-resource-share-create-from-policy` affinché la risorsa sia visibile a tutti i principali con cui è condivisa la risorsa.

```
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" ] } ] }'
```

Output:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/examplecomponent/2019.12.02/1"
```



```
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [PutComponentPolicy](#) in AWS CLI Command Reference.

## put-image-policy

Il seguente esempio di codice mostra come utilizzare `put-image-policy`.

### AWS CLI

Per applicare una politica delle risorse a un'immagine

Il `put-image-policy` comando seguente applica una politica delle risorse a un'immagine per consentire la condivisione delle immagini tra account. Si consiglia di utilizzare il RAM CLI comando `create-resource-share`. Se si utilizza il CLI comando EC2 Image Builder `put-image-policy`, è necessario utilizzare anche il RAM CLI comando `promote-resource-share-create -from-policy` affinché la risorsa sia visibile a tutti i principali con cui è condivisa la risorsa.

```
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" ] } ] }'
```

Output:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imageArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/example-  
image/2019.12.02/1"  
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [PutImagePolicy](#) in AWS CLI Command Reference.

## put-image-recipe-policy

Il seguente esempio di codice mostra come utilizzare `put-image-recipe-policy`.

### AWS CLI

Per applicare una politica delle risorse a una ricetta di immagini

Il `put-image-recipe-policy` comando seguente applica una politica delle risorse a una ricetta di immagini per consentire la condivisione tra account di ricette di immagini. Si consiglia di utilizzare il RAM CLI comando `create-resource-share`. Se si utilizza il CLI comando EC2 Image Builder `put-image-recipe-policy`, è necessario utilizzare anche il RAM CLI comando `promote-resource-share-create-from-policy` affinché la risorsa sia visibile a tutti i principali con cui è condivisa la risorsa.

```
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" ] } ] }'
```

Output:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/  
example-image-recipe/2019.12.02/1"  
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2 Image Builder.

- Per API i dettagli, vedere [PutImageRecipePolicy](#) in AWS CLI Command Reference.

## start-image-pipeline-execution

Il seguente esempio di codice mostra come utilizzare `start-image-pipeline-execution`.

## AWS CLI

Per avviare manualmente una pipeline di immagini

L'`start-image-pipeline-execution`esempio seguente avvia manualmente una pipeline di immagini.

```
aws imagebuilder start-image-pipeline-execution \  
  --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/  
mywindows2016pipeline
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#)utente di EC2Image Builder.

- Per API i dettagli, vedere [StartImagePipelineExecution](#)in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

## AWS CLI

Per etichettare una risorsa

L'`tag-resource`esempio seguente aggiunge e contrassegna una risorsa a EC2 Image Builder utilizzando un JSON file.

```
aws imagebuilder tag-resource \  
  --cli-input-json file://tag-resource.json
```

Contenuto di `tag-resource.json`.

```
{
```

```
"resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
  "tags": {
    "KeyName": "KeyValue"
  }
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da una risorsa

L'`untag-resource` esempio seguente rimuove un tag da una risorsa utilizzando un JSON file.

```
aws imagebuilder untag-resource \
  --cli-input-json file://tag-resource.json
```

Contenuto di `untag-resource.json`.

```
{
  "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
  "tagKeys": [
    "KeyName"
  ]
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-distribution-configuration

Il seguente esempio di codice mostra come utilizzare `update-distribution-configuration`.

### AWS CLI

Per aggiornare una configurazione di distribuzione

L'esempio seguente aggiorna una configurazione di distribuzione utilizzando un JSON file.

```
aws imagebuilder update-distribution-configuration \  
  --cli-input-json file://update-distribution-configuration.json
```

Contenuto di `update-distribution-configuration.json`.

```
{  
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/myexempldistribution",  
  "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}}"  
      }  
    }  
  ]  
}
```

Output:

```
{  
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

```
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l' AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [UpdateDistributionConfiguration](#) in AWS CLI Command Reference.

## update-image-pipeline

Il seguente esempio di codice mostra come utilizzare `update-image-pipeline`.

### AWS CLI

Per aggiornare una pipeline di immagini

L'`update-image-pipeline` esempio seguente aggiorna una pipeline di immagini utilizzando un JSON file.

```
aws imagebuilder update-image-pipeline \  
  --cli-input-json file://update-image-pipeline.json
```

Contenuto di `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"  
}
```

```
}
```

Output:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l'AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [UpdateImagePipeline](#) in AWS CLI Command Reference.

## update-infrastructure-configuration

Il seguente esempio di codice mostra come utilizzare `update-infrastructure-configuration`.

AWS CLI

Per aggiornare una configurazione dell'infrastruttura

L'`update-infrastructure-configuration` esempio seguente aggiorna una configurazione dell'infrastruttura utilizzando un JSON file.

```
aws imagebuilder update-infrastructure-configuration \
  --cli-input-json file:/update-infrastructure-configuration.json
```

Contenuto di `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"
  }
```

Output:

```
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Per ulteriori informazioni, vedere [Configurazione e gestione di una pipeline di EC2 immagini di Image Builder utilizzando la Guida per l'AWS CLI](#) utente di EC2Image Builder.

- Per API i dettagli, vedere [UpdateInfrastructureConfiguration](#) in AWS CLI Command Reference.

## Esempi di Incident Manager che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Incident Manager.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-replication-set**

Il seguente esempio di codice mostra come utilizzare `create-replication-set`.



## AWS CLI

Per creare il set di replica

L'`create-replication-set` seguente crea il set di replica utilizzato da Incident Manager per replicare e crittografare i dati nel tuo account Amazon Web Services. Questo esempio utilizza le regioni `us-east-1` e `us-east-2` durante la creazione del set di replica.

```
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"}}'
```

Output:

```
{  
  "replicationSetArns": [  
    "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-  
bb3f-413c-08df53673b57"  
  ]  
}
```

Per ulteriori informazioni, vedere [Utilizzo del set di replica di Incident Manager](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [CreateReplicationSet](#) in AWS CLI Command Reference.

## `create-response-plan`

Il seguente esempio di codice mostra come utilizzare `create-response-plan`.

## AWS CLI

Per creare un piano di risposta

L'`create-response-plan` seguente crea un piano di risposta con i dettagli specificati.

```
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"]'

```

Output:

```

{
  "arn": "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
}

```

Per ulteriori informazioni, vedere [Preparazione degli incidenti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [CreateResponsePlan](#) in AWS CLI Command Reference.

## create-timeline-event

Il seguente esempio di codice mostra come utilizzare `create-timeline-event`.

AWS CLI

Esempio 1: per creare un evento temporale personalizzato

L'`create-timeline-event` seguente crea un evento temporale personalizzato all'ora specificata dell'incidente specificato.

```

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"

```

Output:

```

{
  "eventId": "c0bcc885-a41d-eb01-b4ab-9d2deEXAMPLE",

```

```
"incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
}
```

Esempio 2: creare un evento cronologico con una nota sull'incidente

L'`create-timeline-event` seguente crea un evento nella sequenza temporale elencato nel pannello «Note sull'incidente».

```
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"]]'
```

Output:

```
{
  "eventId": "a41dc885-c0bc-b4ab-eb01-de9d2EXAMPLE",
  "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
}
```

Per ulteriori informazioni, consulta [i dettagli dell'incidente](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [CreateTimelineEvent](#) in AWS CLI Command Reference.

## **delete-incident-record**

Il seguente esempio di codice mostra come utilizzare `delete-incident-record`.

### AWS CLI

Per eliminare un record di incidente

L'`delete-incident-record` seguente elimina il record di incidente specificato.

```
aws ssm-incidents delete-incident-record \
```

```
--arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Incident tracking](#) nella Incident Manager User Guide.

- Per API i dettagli, vedere [DeleteIncidentRecord](#) in AWS CLI Command Reference.

## **delete-replication-set**

Il seguente esempio di codice mostra come utilizzare `delete-replication-set`.

### AWS CLI

Per eliminare il set di replica

L'`delete-replication-set` seguente elimina il set di repliche dal tuo account Amazon Web Services. L'eliminazione del set di repliche elimina anche tutti i dati di Incident Manager. Questa operazione non può essere annullata.

```
aws ssm-incidents delete-replication-set \  
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-  
  bb3f-413c-08df53673b57"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Utilizzo del set di replica di Incident Manager](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [DeleteReplicationSet](#) in AWS CLI Command Reference.

## **delete-resource-policy**

Il seguente esempio di codice mostra come utilizzare `delete-resource-policy`.

### AWS CLI

Per eliminare una politica delle risorse

L'`delete-resource-policy` seguente elimina una politica delle risorse da un piano di risposta. Ciò revocherà l'accesso al principale o all'organizzazione con cui è stato condiviso il piano di risposta.

```
aws ssm-incidents delete-resource-policy \  
  --policy-id "be8b57191f0371f1c6827341aa3f0a03" \  
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Utilizzo dei contatti condivisi e dei piani di risposta](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [DeleteResourcePolicy](#) in AWS CLI Command Reference.

## delete-response-plan

Il seguente esempio di codice mostra come utilizzare `delete-response-plan`.

### AWS CLI

Per eliminare un piano di risposta

L'esempio seguente elimina il piano di risposta specificato.

```
aws ssm-incidents delete-response-plan \  
  --arn "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Preparazione degli incidenti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [DeleteResponsePlan](#) in AWS CLI Command Reference.

## delete-timeline-event

Il seguente esempio di codice mostra come utilizzare `delete-timeline-event`.

### AWS CLI

Per eliminare un evento della timeline

L'esempio seguente elimina un evento cronologico personalizzato dal record dell'incidente specificato.

```
aws ssm-incidents delete-timeline-event \  
  --event-id "c0bcc885-a41d-eb01-b4ab-9d2de193643c" \  
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [i dettagli dell'incidente nella Guida](#) per l'utente di Incident Manager.

- Per API i dettagli, vedere [DeleteTimelineEvent](#) in AWS CLI Command Reference.

## get-incident-record

Il seguente esempio di codice mostra come utilizzare `get-incident-record`.

### AWS CLI

Per ottenere un registro degli incidenti

L'`get-incident-record` seguente ottiene dettagli sul record di incidente specificato.

```
aws ssm-incidents get-incident-record \  
  --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-  
Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, vedere [i dettagli dell'incidente](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [GetIncidentRecord](#) in AWS CLI Command Reference.

## get-replication-set

Il seguente esempio di codice mostra come utilizzare `get-replication-set`.

### AWS CLI

Per ottenere il set di replica

L'`get-replication-set` seguente ottiene i dettagli del set di replica utilizzato da Incident Manager per replicare e crittografare i dati nel tuo account Amazon Web Services.

```

aws ssm-incidents get-replication-set \
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
bb3f-413c-08df53673b57"

```

Output:

```

{
  "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"
}

```

Per ulteriori informazioni, consulta [Using the Incident Manager replication set](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [GetReplicationSet](#) in AWS CLI Command Reference.

## get-resource-policies

Il seguente esempio di codice mostra come utilizzare `get-resource-policies`.

### AWS CLI

Per elencare le politiche relative alle risorse per un piano di risposta

L'command-name esempio seguente elenca le politiche relative alle risorse associate al piano di risposta specificato.

```

aws ssm-incidents get-resource-policies \
--resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"

```

Output:

```

{
  "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\"
  }
]
}

```

Per ulteriori informazioni, vedere [Utilizzo dei contatti condivisi e dei piani di risposta](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [GetResourcePolicies](#) in AWS CLI Command Reference.

## get-response-plan

Il seguente esempio di codice mostra come utilizzare `get-response-plan`.

### AWS CLI

Per ottenere i dettagli di un piano di risposta

L'command-nameesempio seguente fornisce dettagli su un piano di risposta specifico nel tuo AWS account.

```

aws ssm-incidents get-response-plan \
  --arn \"arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan\"

```

Output:

```

{
  \"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"
}

```

Per ulteriori informazioni, consulta la sezione [Preparazione degli incidenti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [GetResponsePlan](#) in AWS CLI Command Reference.

## get-timeline-event

Il seguente esempio di codice mostra come utilizzare `get-timeline-event`.

### AWS CLI

Per ottenere i dettagli di un evento nella sequenza temporale

L'`get-timeline-event` esempio seguente restituisce i dettagli dell'evento temporale specificato.

```

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"

```

Output:

```

{
  "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"
  }
}
```

Per ulteriori informazioni, vedere [i dettagli degli incidenti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [GetTimelineEvent](#) in AWS CLI Command Reference.

## list-incident-records

Il seguente esempio di codice mostra come utilizzare `list-incident-records`.

### AWS CLI

Per elencare i record degli incidenti

L'command-name esempio seguente elenca i record degli incidenti nel tuo account Amazon Web Services.

```
aws ssm-incidents list-incident-records
```

Output:

```
{
  "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/draliatp",
        "invokedBy": "arn:aws:iam::111122223333:user/draliatp",
        "source": "aws.ssm-incidents.custom"
      },
      "status": "OPEN",
      "title": "Example-Incident"
    }
  ]
}
```

```
    }
  ]
}
```

Per ulteriori informazioni, consulta [l'elenco degli incidenti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [ListIncidentRecords](#) in AWS CLI Command Reference.

## list-related-items

Il seguente esempio di codice mostra come utilizzare `list-related-items`.

### AWS CLI

Per elencare gli articoli correlati

L'`list-related-items` esempio seguente elenca gli elementi correlati dell'incidente specificato.

```
aws ssm-incidents list-related-items \
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
  Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [i dettagli dell'incidente](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [ListRelatedItems](#) in AWS CLI Command Reference.

## list-replication-sets

Il seguente esempio di codice mostra come utilizzare `list-replication-sets`.

### AWS CLI

Per elencare il set di replica

L'`list-replication-sets` esempio seguente elenca il set di replica utilizzato da Incident Manager per replicare e crittografare i dati nell'account. AWS

```
aws ssm-incidents list-replication-sets
```

Output:

```
{
  "replicationSetArns": [
    "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
    bb3f-413c-08df53673b57"
  ]
}
```

Per ulteriori informazioni, vedere [Utilizzo del set di replica di Incident Manager](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [ListReplicationSets](#) in AWS CLI Command Reference.

## list-response-plans

Il seguente esempio di codice mostra come utilizzare `list-response-plans`.

## AWS CLI

Per elencare i piani di risposta disponibili

L'`list-response-plans` seguente elenca i piani di risposta disponibili nel tuo account Amazon Web Services.

```
aws ssm-incidents list-response-plans
```

Output:

```
{
  "responsePlanSummaries": [
    {
      "arn": "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan",
      "displayName": "Example response plan",
      "name": "Example-Response-Plan"
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione [Preparazione degli incidenti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [ListResponsePlans](#) in AWS CLI Command Reference.

## `list-tags-for-resource`

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

## AWS CLI

Per elencare i tag per un piano di risposta

L'`list-tags-for-resource` seguente elenca i tag associati al piano di risposta specificato.

```
aws ssm-incidents list-tags-for-resource \
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

**Output:**

```
{
  "tags": {
    "group1": "1"
  }
}
```

Per ulteriori informazioni, vedere [Tagging](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

**list-timeline-events**

Il seguente esempio di codice mostra come utilizzare `list-timeline-events`.

**AWS CLI**

Per elencare gli eventi cronologici di un incidente

L'`command-name`esempio seguente elenca gli eventi cronologici dell'incidente specificato.

```
aws ssm-incidents list-timeline-events \
  --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
  Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

**Output:**

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, vedere [i dettagli degli incidenti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [ListTimelineEvents](#) in AWS CLI Command Reference.



## put-resource-policy

Il seguente esempio di codice mostra come utilizzare `put-resource-policy`.

### AWS CLI

Per condividere un piano di risposta e gli incidenti

L'command-nameesempio seguente aggiunge una politica delle risorse a `Example-Response-Plan` quella che condivide il piano di risposta e gli incidenti associati con il principale specificato.

```
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/*\"]}]}"
```

Output:

```
{
  "policyId": "be8b57191f0371f1c6827341aa3f0a03"
}
```

Per ulteriori informazioni, vedere [Utilizzo dei contatti condivisi e dei piani di risposta](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [PutResourcePolicy](#) in AWS CLI Command Reference.

## start-incident

Il seguente esempio di codice mostra come utilizzare `start-incident`.

### AWS CLI

Per iniziare un incidente

L'`start-incident` seguente avvia un incidente utilizzando il piano di risposta specificato.

```
aws ssm-incidents start-incident \  
  --response-plan-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

Output:

```
{  
  "incidentRecordArn": "arn:aws:ssm-incidents::682428703967:incident-record/  
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"  
}
```

Per ulteriori informazioni, vedere [Creazione di incidenti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [StartIncident](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

AWS CLI

Per etichettare un piano di risposta

L'`tag-resource` seguente contrassegna un piano di risposta specificato con la coppia chiave-valore del tag fornita.

```
aws ssm-incidents tag-resource \  
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan" \  
  --tags '{"group1":"1"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Tagging](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da un piano di risposta

L'`untag-resource` esempio seguente rimuove i tag specificati dal piano di risposta.

```
aws ssm-incidents untag-resource \  
  --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-  
Response-Plan" \  
  --tag-keys ["group1"]
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Tagging](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-deletion-protection

Il seguente esempio di codice mostra come utilizzare `update-deletion-protection`.

### AWS CLI

Per aggiornare la protezione dall'eliminazione dei set di replica

L'`update-deletion-protection` esempio seguente aggiorna la protezione dall'eliminazione dell'account per impedire l'eliminazione dell'ultima regione del set di repliche.

```
aws ssm-incidents update-deletion-protection \  
  --arn "arn:aws:ssm-incidents::111122223333:replication-set/  
a2bcc5c9-0f53-8047-7fef-c20749989b40" \  
  --deletion-protected
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Utilizzo del set di replica di Incident Manager](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [UpdateDeletionProtection](#) in AWS CLI Command Reference.

## update-incident-record

Il seguente esempio di codice mostra come utilizzare `update-incident-record`.

### AWS CLI

Per aggiornare il record di un incidente

L'command-nameesempio seguente risolve l'incidente specificato.

```
aws ssm-incidents update-incident-record \
  --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \
  --status "RESOLVED"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [i dettagli dell'incidente nella Guida](#) per l'utente di Incident Manager.

- Per API i dettagli, vedere [UpdateIncidentRecord](#) in AWS CLI Command Reference.

## update-related-items

Il seguente esempio di codice mostra come utilizzare `update-related-items`.

### AWS CLI

Per aggiornare un elemento relativo agli incidenti

L'update-related-itemesempio seguente rimuove un elemento correlato dal record dell'incidente specificato.

```
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"}}}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [i dettagli dell'incidente](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [UpdateRelatedItems](#) in AWS CLI Command Reference.

## update-replication-set

Il seguente esempio di codice mostra come utilizzare `update-replication-set`.

### AWS CLI

Per aggiornare un set di replica

L'`command-name`esempio seguente elimina la regione `us-east-2` dal set di replica.

```
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"}}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Utilizzo del set di replica di Incident Manager](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [UpdateReplicationSet](#) in AWS CLI Command Reference.

## update-response-plan

Il seguente esempio di codice mostra come utilizzare `update-response-plan`.

### AWS CLI

Per aggiornare un piano di risposta

L'`update-response-plan`esempio seguente rimuove un canale di chat dal piano di risposta specificato.

```
aws ssm-incidents update-response-plan \  
  --arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"  
 \  
  --chat-channel '{"empty":{}}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Preparazione degli incidenti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [UpdateResponsePlan](#) in AWS CLI Command Reference.

## update-timeline-event

Il seguente esempio di codice mostra come utilizzare `update-timeline-event`.

### AWS CLI

Per aggiornare un evento della timeline

L'`update-timeline-event` seguente aggiorna l'ora in cui si è verificato l'evento.

```
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"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [i dettagli dell'incidente](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [UpdateTimelineEvent](#) in AWS CLI Command Reference.

## Esempi di contatti di Incident Manager utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Incident Manager Contacts.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto degli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **accept-page**

Il seguente esempio di codice mostra come utilizzare `accept-page`.

#### AWS CLI

Accettare una pagina durante un coinvolgimento

L'esempio seguente utilizza un codice di accettazione inviato al canale di contatto per accettare una pagina.

```
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
```

Questo comando non produce alcun output

Per ulteriori informazioni, vedere [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [AcceptPage](#) in AWS CLI Command Reference.

### **activate-contact-channel**

Il seguente esempio di codice mostra come utilizzare `activate-contact-channel`.

#### AWS CLI

Attiva il canale di contatto di un contatto

L'esempio seguente attiva un canale di contatto e lo rende utilizzabile come parte di un incidente.

```
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"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [ActivateContactChannel](#) in AWS CLI Command Reference.

## command-name

Il seguente esempio di codice mostra come utilizzare `command-name`.

### AWS CLI

Per eliminare un contatto

L'`command-name` esempio seguente elimina un contatto. Il contatto non sarà più raggiungibile da nessun piano di escalation che lo riguarda.

```
aws ssm-contacts delete-contact \  
  --contact-id "arn:aws:ssm-contacts:us-east-1:682428703967:contact/alejr"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [CommandName](#) in AWS CLI Command Reference.

## create-contact-channel

Il seguente esempio di codice mostra come utilizzare `create-contact-channel`.

### AWS CLI

Per creare un canale di contatto

Crea un canale di contatto di tipo SMS per il contatto Akua Mansa. I canali di contatto possono essere creati di tipo SMSEMAIL, o. 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
```



```
--type SMS \  
--delivery-address '{"SimpleAddress": "+15005550199"}'
```

Output:

```
{  
  "ContactChannelArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact-  
channel/akuam/02f506b9-ea5d-4764-af89-2daa793ff024"  
}
```

Per ulteriori informazioni, consulta [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [CreateContactChannel](#) in AWS CLI Command Reference.

## create-contact

Il seguente esempio di codice mostra come utilizzare `create-contact`.

### AWS CLI

Per creare un contatto

L'esempio seguente crea un contatto nell'ambiente in uso con un piano vuoto. Il piano può essere aggiornato dopo aver creato i canali di contatto. Usa il `create-contact-channel` comando con l'output ARN di questo comando. Dopo aver creato i canali di contatto per questo contatto, usa `update-contact` per aggiornare il piano.

```
aws ssm-contacts create-contact \  
  --alias "akuam" \  
  --display-name "Akua Mansa" \  
  --type PERSONAL \  
  --plan '{"Stages": []}'
```

Output:

```
{  
  "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"  
}
```

Per ulteriori informazioni, consulta [Contatti nella Guida per l'utente di Incident Manager](#).

- Per API i dettagli, vedere [CreateContact](#) in AWS CLI Command Reference.

## **deactivate-contact-channel**

Il seguente esempio di codice mostra come utilizzare `deactivate-contact-channel`.

### AWS CLI

Per disattivare un canale di contatto

L'esempio seguente disattiva un canale di contatto. La disattivazione di un canale di contatto significa che il canale di contatto non verrà più contattato durante un incidente. Puoi anche riattivare un canale di contatto in qualsiasi momento utilizzando il comando `activate-contact-channel`.

```
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"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [DeactivateContactChannel](#) in AWS CLI Command Reference.

## **delete-contact-channel**

Il seguente esempio di codice mostra come utilizzare `delete-contact-channel`.

### AWS CLI

Per eliminare un canale di contatto

L'esempio seguente elimina un canale di contatto. L'eliminazione di un canale di contatto garantisce che il canale di contatto non venga contattato durante un incidente.

```
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"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Contatti nella Guida per l'utente](#) di Incident Manager.

- Per API i dettagli, vedere [DeleteContactChannel](#) in AWS CLI Command Reference.

## delete-contact

Il seguente esempio di codice mostra come utilizzare `delete-contact`.

### AWS CLI

Per eliminare un contatto

L'`delete-contact` esempio seguente elimina un contatto. Il contatto non sarà più raggiungibile da nessun piano di escalation che lo riguardi.

```
aws ssm-contacts delete-contact \  
  --contact-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact/alejr"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [DeleteContact](#) in AWS CLI Command Reference.

## describe-engagement

Il seguente esempio di codice mostra come utilizzare `describe-engagement`.

### AWS CLI

Descrivere i dettagli di un impegno

L'`describe-engagement` esempio seguente elenca i dettagli di un coinvolgimento in un piano di contatto o di escalation. L'oggetto e il contenuto vengono inviati ai canali di contatto.

```
aws ssm-contacts describe-engagement \  
  --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
  example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

Output:

```
{
  "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-4d57-8962-5fbe7fc53356",
  "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"
}
```

Per ulteriori informazioni, consulta [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [DescribeEngagement](#) in AWS CLI Command Reference.

## describe-page

Il seguente esempio di codice mostra come utilizzare `describe-page`.

### AWS CLI

Per elencare i dettagli di una pagina in un canale di contatto

L'`describe-page` esempio seguente elenca i dettagli di una pagina in un canale di contatto. La pagina includerà l'oggetto e il contenuto forniti.

```
aws ssm-contacts describe-page \
  --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/ad0052bd-
e606-498a-861b-25726292eb93"
```

Output:

```
{
  "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"
}

```

Per ulteriori informazioni, consulta [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [DescribePage](#) in AWS CLI Command Reference.

## get-contact-channel

Il seguente esempio di codice mostra come utilizzare `get-contact-channel`.

### AWS CLI

Per elencare i dettagli di un canale di contatto

L'`get-contact-channel` esempio seguente elenca i dettagli di un canale di contatto.

```

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"

```

Output:

```

{
  "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"
}

```

Per ulteriori informazioni, vedere [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [GetContactChannel](#) in AWS CLI Command Reference.

## get-contact-policy

Il seguente esempio di codice mostra come utilizzare `get-contact-policy`.

### AWS CLI

Per elencare le politiche relative alle risorse di un contatto

L'`get-contact-policy` esempio seguente elenca le politiche relative alle risorse associate al contatto specificato.

```
aws ssm-contacts get-contact-policy \
  --contact-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

Output:

```
{
  "ContactArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam",
  "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
  \\\"SharePolicyForDocumentationDralia\\\",\\\"Effect\\\":\\\"Allow\\\",\\\"Principal\\\":
  {\\\"AWS\\\":\\\"222233334444\\\"},\\\"Action\\\":[\\\"ssm-contacts:GetContact\\\",\\\"ssm-
  contacts:StartEngagement\\\",\\\"ssm-contacts:DescribeEngagement\\\",\\\"ssm-
  contacts:ListPagesByEngagement\\\",\\\"ssm-contacts:StopEngagement\\\"],\\\"Resource
  \\\":[\\\"arn:aws:ssm-contacts:*:111122223333:contact/akuam\\\",\\\"arn:aws:ssm-
  contacts:*:111122223333:engagement/akuam/*\\\"]}]}"
}
```

Per ulteriori informazioni, vedere [Utilizzo dei contatti condivisi e dei piani di risposta](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [GetContactPolicy](#) in AWS CLI Command Reference.

## get-contact

Il seguente esempio di codice mostra come utilizzare `get-contact`.

### AWS CLI

Esempio 1: Per descrivere un piano di contatto

L'get-contactesempio seguente descrive un contatto.

```
aws ssm-contacts get-contact \  
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

Output:

```
{  
  "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
    }
}

```

## Esempio 2: descrivere un piano di escalation

L'get-contactesempio seguente descrive un piano di escalation.

```

aws ssm-contacts get-contact \
--contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation"

```

Output:

```

{
  "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
                    }
                }
            ]
        }
    ]
}

```

Per ulteriori informazioni, vedere [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [GetContact](#) in AWS CLI Command Reference.

## list-contact-channels

Il seguente esempio di codice mostra come utilizzare `list-contact-channels`.

### AWS CLI

Per elencare i canali di contatto di un contatto

L'esempio seguente elenca i canali di contatto disponibili del contatto specificato.

```

aws ssm-contacts list-contact-channels \
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"

```

**Output:**

```
{
  [
    {
      "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [ListContactChannels](#) in AWS CLI Command Reference.

**list-contacts**

Il seguente esempio di codice mostra come utilizzare `list-contacts`.

**AWS CLI**

Per elencare tutti i piani di escalation e i contatti

L'`list-contacts` esempio seguente elenca i contatti e i piani di escalation presenti nel tuo account.

```
aws ssm-contacts list-contacts
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Contatti nella Guida per l'utente di Incident Manager](#).

- Per API i dettagli, vedere [ListContacts](#) in AWS CLI Command Reference.

## list-engagements

Il seguente esempio di codice mostra come utilizzare `list-engagements`.

### AWS CLI

Per elencare tutti gli impegni

L'`list-engagements` seguente elenca gli impegni relativi ai piani di escalation e ai contatti. È inoltre possibile elencare gli impegni per un singolo incidente.

```
aws ssm-contacts list-engagements
```

Output:

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Contatti nella Guida per l'utente](#) di Incident Manager.

- Per API i dettagli, vedere [ListEngagements](#) in AWS CLI Command Reference.

## list-page-receipts

Il seguente esempio di codice mostra come utilizzare `list-page-receipts`.

### AWS CLI

Per elencare le ricevute della pagina

L'command-name esempio seguente elenca se una pagina è stata ricevuta o meno da un contatto.

```

aws ssm-contacts list-page-receipts \
  --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/
akuam/94ea0c7b-56d9-46c3-b84a-a37c8b067ad3"

```

Output:

```

{
  "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"
    }
]
}

```

Per ulteriori informazioni, vedere [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [ListPageReceipts](#) in AWS CLI Command Reference.

## list-pages-by-contact

Il seguente esempio di codice mostra come utilizzare `list-pages-by-contact`.

### AWS CLI

Per elencare le pagine per contatto

L'`list-pages-by-contact` esempio seguente elenca tutte le pagine relative al contatto specificato.

```

aws ssm-contacts list-pages-by-contact \
  --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"

```

Output:

```

{
  "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"
    }
]
}

```

Per ulteriori informazioni, vedere [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [ListPagesByContact](#) in AWS CLI Command Reference.

## list-pages-by-engagement

Il seguente esempio di codice mostra come utilizzare `list-pages-by-engagement`.

### AWS CLI

Elencare le pagine dei canali di contatto partendo da un engagement.

L'`list-pages-by-engagement` esempio seguente elenca le pagine che si sono verificate durante l'implementazione del piano di coinvolgimento definito.

```

aws ssm-contacts list-pages-by-engagement \
  --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
  akuam/78a29753-3674-4ac5-9f83-0468563567f0"

```

### Output:

```

{
  "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"
    }
  ]
}

```

Per ulteriori informazioni, vedere [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [ListPagesByEngagement](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di un contatto

L'`list-tags-for-resource` esempio seguente elenca i tag del contatto specificato.

```
aws ssm-contacts list-tags-for-resource \  
  --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

Output:

```
{  
  "Tags": [  
    {  
      "Key": "group1",  
      "Value": "1"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Tagging](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## put-contact-policy

Il seguente esempio di codice mostra come utilizzare `put-contact-policy`.

### AWS CLI

Per condividere un contatto e gli impegni

L'`put-contact-policy` esempio seguente aggiunge una politica sulle risorse al contatto Akua che condivide il contatto e gli impegni correlati con il principale.



```
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/*\"]}]}"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Utilizzo dei contatti condivisi e dei piani di risposta](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [PutContactPolicy](#) in AWS CLI Command Reference.

## send-activation-code

Il seguente esempio di codice mostra come utilizzare `send-activation-code`.

### AWS CLI

Per inviare un codice di attivazione

L'esempio seguente invia un codice di attivazione e un messaggio al canale di contatto specificato.

```
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"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [SendActivationCode](#) in AWS CLI Command Reference.

## start-engagement

Il seguente esempio di codice mostra come utilizzare `start-engagement`.

## AWS CLI

### Esempio 1: Pagine i canali di contatto di un contatto

Le `start-engagement` pagine seguenti: canali di contatto per i contatti. Il mittente, l'oggetto, l'oggetto pubblico e il contenuto pubblico sono tutti privi di campi. Incident Manager invia l'oggetto e il contenuto ai canali forniti o di contatto. VOICE EMAIL Incident Manager invia l'argomento pubblico e il contenuto pubblico ai canali di contatto forniti. SMS Il mittente viene utilizzato per tenere traccia di chi ha avviato il coinvolgimento.

```
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"
```

Output:

```
{  
  "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/  
akuam/607ced0e-e8fa-4ea7-8958-a237b8803f8f"  
}
```

Per ulteriori informazioni, consulta [Contatti](#) nella Guida per l'utente di Incident Manager.

Esempio 2: inserire un contatto nel piano di escalation fornito.

Quanto segue `start-engagement` coinvolge i contatti attraverso un piano di escalation. Ogni contatto viene suddiviso in base al proprio piano di coinvolgimento.

```
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"
```

## Output:

```
{
  "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
}
```

Per ulteriori informazioni, consulta [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [StartEngagement](#) in AWS CLI Command Reference.

## stop-engagement

Il seguente esempio di codice mostra come utilizzare `stop-engagement`.

### AWS CLI

Interrompere un fidanzamento

L'`stop-engagement` esempio seguente impedisce a un coinvolgimento di cercare ulteriori contatti e canali di contatto.

```
aws ssm-contacts stop-engagement \
  --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [StopEngagement](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per taggare un contatto

L'`tag-resource` esempio seguente contrassegna un contatto specificato con la coppia chiave-valore del tag fornita.

```
aws ssm-contacts tag-resource \  
  --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \  
  --tags '[{"Key":"group1","Value":"1"}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Tagging](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da un contatto

L'`untag-resource` esempio seguente rimuove il tag `group1` dal contatto specificato.

```
aws ssm-contacts untag-resource \  
  --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \  
  --tag-keys "group1"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Tagging](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-contact-channel

Il seguente esempio di codice mostra come utilizzare `update-contact-channel`.

### AWS CLI

Per aggiornare un canale di contatto

L'`update-contact-channel` esempio seguente aggiorna il nome e l'indirizzo di consegna di un canale di contatto.

```
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"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [UpdateContactChannel](#) in AWS CLI Command Reference.

## update-contact

Il seguente esempio di codice mostra come utilizzare `update-contact`.

### AWS CLI

Per aggiornare il piano di coinvolgimento dei contatti

L'`update-contact` esempio seguente aggiorna il piano di coinvolgimento del contatto Akua per includere i tre tipi di canali di contatto. Questa operazione viene eseguita dopo aver creato i canali di contatto per 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 }]}]}]}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Contatti](#) nella Guida per l'utente di Incident Manager.

- Per API i dettagli, vedere [UpdateContact](#) in AWS CLI Command Reference.

## Esempi di utilizzo di Amazon Inspector AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Inspector. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **add-attributes-to-findings**

Il seguente esempio di codice mostra come utilizzare `add-attributes-to-findings`.

AWS CLI

Per aggiungere attributi ai risultati

Il `add-attribute-to-finding` comando seguente assegna un attributo con la chiave `Example` e il valore di `example` al risultato con ARN `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-811VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU`:

```
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
```

Output:

```
{
  "failedItems": {}
}
```

Per ulteriori informazioni, consulta Amazon Inspector Findings nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [AddAttributesToFindingsReference](#).

## create-assessment-target

Il seguente esempio di codice mostra come utilizzare `create-assessment-target`.

### AWS CLI

Per creare un obiettivo di valutazione

Il `create-assessment-target` comando seguente crea un obiettivo di valutazione denominato `ExampleAssessmentTarget` utilizzando il gruppo ARN di risorse `arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-AB6DMKnv`:

```
aws inspector create-assessment-target --assessment-target-name ExampleAssessmentTarget --resource-group-arn arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-AB6DMKnv
```

Output:

```
{
  "assessmentTargetArn": "arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX"
}
```

Per ulteriori informazioni, consulta gli obiettivi di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [CreateAssessmentTargetReference](#).

## create-assessment-template

Il seguente esempio di codice mostra come utilizzare `create-assessment-template`.

### AWS CLI

Per creare un modello di valutazione

Il `create-assessment-template` comando seguente crea un modello di valutazione chiamato `ExampleAssessmentTemplate` per l'obiettivo ARN di valutazione `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
```

Output:

```
{
  "assessmentTemplateArn": "arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T"
}
```

Per ulteriori informazioni, consulta i modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [CreateAssessmentTemplate](#) Reference.

## create-filter

Il seguente esempio di codice mostra come utilizzare `create-filter`.

### AWS CLI

Per creare un filtro

L'`create-filter` esempio seguente crea una regola di soppressione che omette i risultati del tipo di ECR istanza.

```
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"}]'
```

Output:

```
{
  "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE444444444444"
```



```
}
```

Per ulteriori informazioni, consulta [Filtrare i risultati di Amazon Inspector](#) nella Amazon Inspector User Guide.

- Per API i dettagli, consulta Command [CreateFilter](#)Reference AWS CLI .

## create-findings-report

Il seguente esempio di codice mostra come utilizzare `create-findings-report`.

### AWS CLI

Per creare un rapporto sui risultati

L'`create-findings-report`esempio seguente crea un rapporto di ricerca.

```
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"}]}'
```

Output:

```
{  
  "reportId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"  
}
```

Per ulteriori informazioni, consulta la sezione [Gestione dei risultati in Amazon Inspector](#) nella Amazon Inspector User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateFindingsReport](#)Reference.

## create-resource-group

Il seguente esempio di codice mostra come utilizzare `create-resource-group`.

## AWS CLI

Per creare un gruppo di risorse

Il `create-resource-group` comando seguente crea un gruppo di risorse utilizzando la chiave `tag Name` e il valore di `example`:

```
aws inspector create-resource-group --resource-group-tags key=Name,value=example
```

Output:

```
{
  "resourceGroupArn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-AB6DMKnv"
}
```

Per ulteriori informazioni, consulta gli obiettivi di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [CreateResourceGroup](#) Reference.

## `create-sbom-export`

Il seguente esempio di codice mostra come utilizzare `create-sbom-export`.

## AWS CLI

Per creare un rapporto sulla distinta base del software (SBOM)

L'`create-sbom-export` esempio seguente crea un rapporto sulla distinta base del software (SBOM).

```
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
```

Output:

```
{
  "reportId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
}
```

Per ulteriori informazioni, consulta [Exporting SBOMs with Amazon Inspector nella Amazon Inspector User Guide](#).

- Per API i dettagli, consulta Command [CreateSbomExport](#)Reference AWS CLI .

## **delete-assessment-run**

Il seguente esempio di codice mostra come utilizzare `delete-assessment-run`.

### AWS CLI

Per eliminare un'esecuzione di valutazione

Il `delete-assessment-run` comando seguente elimina l'esecuzione di valutazione con ARN `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-11LMTAVe`:

```
aws inspector delete-assessment-run --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-11LMTAVe
```

Per ulteriori informazioni, consulta i modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [DeleteAssessmentRun](#)Reference.

## **delete-assessment-target**

Il seguente esempio di codice mostra come utilizzare `delete-assessment-target`.

### AWS CLI

Per eliminare un target di valutazione

Il `delete-assessment-target` comando seguente elimina l'obiettivo di valutazione con il comando ARN `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
```

Per ulteriori informazioni, consulta gli obiettivi di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [DeleteAssessmentTarget](#)Reference.

## delete-assessment-template

Il seguente esempio di codice mostra come utilizzare `delete-assessment-template`.

### AWS CLI

Per eliminare un modello di valutazione

Il `delete-assessment-template` comando seguente elimina il modello di valutazione con il comando ARN of `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T`:

```
aws inspector delete-assessment-template --assessment-template-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T
```

Per ulteriori informazioni, consulta i modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [DeleteAssessmentTemplate](#)Reference.

## delete-filter

Il seguente esempio di codice mostra come utilizzare `delete-filter`.

### AWS CLI

Per eliminare un filtro

L'`delete-filter` esempio seguente elimina un filtro.

```
aws inspector2 delete-filter \  
  --arn "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE444444444"
```

**Output:**

```
{
  "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/EXAMPLE4444444444"
}
```

Per ulteriori informazioni, consulta [Filtrare i risultati di Amazon Inspector](#) nella Amazon Inspector User Guide.

- Per API i dettagli, consulta Command [DeleteFilter](#)Reference AWS CLI .

**describe-assessment-runs**

Il seguente esempio di codice mostra come utilizzare `describe-assessment-runs`.

**AWS CLI**

Per descrivere le esecuzioni di valutazione

Il `describe-assessment-run` comando seguente descrive un'esecuzione di valutazione con ARN `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
```

**Output:**

```
{
  "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": {}
```

```
}
```

Per ulteriori informazioni, consulta i modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [DescribeAssessmentRuns](#) Reference.

## describe-assessment-targets

Il seguente esempio di codice mostra come utilizzare `describe-assessment-targets`.

### AWS CLI

Descrivere gli obiettivi di valutazione

Il `describe-assessment-targets` comando seguente descrive l'obiettivo ARN di valutazione con `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
```

Output:

```
{
  "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": {}
}
```

Per ulteriori informazioni, consulta gli obiettivi di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [DescribeAssessmentTargets](#) Reference.

## describe-assessment-templates

Il seguente esempio di codice mostra come utilizzare `describe-assessment-templates`.

### AWS CLI

Per descrivere i modelli di valutazione

Il `describe-assessment-templates` comando seguente descrive il modello ARN di valutazione con `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
```

Output:

```
{
  "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": {}
}
```

Per ulteriori informazioni, consulta i modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [DescribeAssessmentTemplates](#) Reference.



## describe-cross-account-access-role

Il seguente esempio di codice mostra come utilizzare `describe-cross-account-access-role`.

### AWS CLI

Per descrivere il ruolo di accesso tra account

Il `describe-cross-account-access-role` comando seguente descrive il IAM ruolo che consente ad Amazon Inspector di accedere al tuo AWS account:

```
aws inspector describe-cross-account-access-role
```

Output:

```
{
  "registeredAt": 1458069182.826,
  "roleArn": "arn:aws:iam::123456789012:role/inspector",
  "valid": true
}
```

Per ulteriori informazioni, consulta Configurazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [DescribeCrossAccountAccessRole](#) Reference.

## describe-findings

Il seguente esempio di codice mostra come utilizzare `describe-findings`.

### AWS CLI

Per descrivere i risultati

Il `describe-findings` comando seguente descrive il risultato con il comando 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
```

## Output:

```
{
  "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": []
    }
  ]
}
```

Per ulteriori informazioni, consulta Amazon Inspector Findings nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [DescribeFindings](#)Reference.

## describe-resource-groups

Il seguente esempio di codice mostra come utilizzare `describe-resource-groups`.

### AWS CLI

Per descrivere i gruppi di risorse

Il `describe-resource-groups` comando seguente descrive il gruppo ARN di risorse con `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
```

Output:

```
{
  "failedItems": {},
  "resourceGroups": [
    {
      "arn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-PyGXopAI",
      "createdAt": 1458074191.098,
      "tags": [
        {
          "key": "Name",
          "value": "example"
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, consulta gli obiettivi di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [DescribeResourceGroups](#) Reference.

## describe-rules-packages

Il seguente esempio di codice mostra come utilizzare `describe-rules-packages`.

## AWS CLI

Per descrivere i pacchetti di regole

Il `describe-rules-packages` comando seguente descrive il pacchetto di regole con 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
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta Amazon Inspector Rules Packages and Rules nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [DescribeRulesPackages](#) Reference.

## get-configuration

Il seguente esempio di codice mostra come utilizzare `get-configuration`.

### AWS CLI

Per ottenere la configurazione delle impostazioni per le scansioni di Inspector

L'`get-configuration` seguente ottiene la configurazione delle impostazioni per le scansioni di Inspector.

```
aws inspector2 get-configuration
```

Output:

```
{
  "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"
    }
  }
}
```

Per ulteriori informazioni, consulta [Scansione automatizzata delle risorse con Amazon Inspector](#) nella Amazon Inspector User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetConfiguration](#) Reference.

## get-telemetry-metadata

Il seguente esempio di codice mostra come utilizzare `get-telemetry-metadata`.

## AWS CLI

Per ottenere i metadati di telemetria

Il `get-telemetry-metadata` comando seguente genera informazioni sui dati raccolti per la valutazione eseguita con of: ARN `arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE`

```
aws inspector get-telemetry-metadata --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-4r1V2mAw/run/0-MKkpXXPE
```

Output:

```
{
  "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"  
  }  
]  
}
```

- Per API i dettagli, vedere [GetTelemetryMetadata](#) in AWS CLI Command Reference.

## list-account-permissions

Il seguente esempio di codice mostra come utilizzare `list-account-permissions`.

### AWS CLI

Per elencare le autorizzazioni dell'account

L'`list-account-permissions` esempio seguente elenca le autorizzazioni dell'account.

**aws inspector2 list-account-permissions**

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Identity and Access Management for Amazon Inspector nella Amazon Inspector User Guide](#).

- Per API i dettagli, consulta AWS CLI Command [ListAccountPermissions](#) Reference.

## list-assessment-run-agents

Il seguente esempio di codice mostra come utilizzare `list-assessment-run-agents`.

### AWS CLI

Per elencare gli agenti di valutazione, esegui

Il `list-assessment-run-agents` comando seguente elenca gli agenti della valutazione eseguita con quanto specificato 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
```

Output:

```
{  
  "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": 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"
}
```

```
}
  ]
}
]
```

Per ulteriori informazioni, consulta [AWS Agenti](#) nella Guida per l'utente di Amazon Inspector.

- Per API i dettagli, consulta [ListAssessmentRunAgents AWS CLI Command Reference](#).

## **list-assessment-runs**

Il seguente esempio di codice mostra come utilizzare `list-assessment-runs`.

### AWS CLI

Per elencare le esecuzioni di valutazione

Il `list-assessment-runs` comando seguente elenca tutte le esecuzioni di valutazione esistenti.

```
aws inspector list-assessment-runs
```

Output:

```
{
  "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"
  ]
}
```

Per ulteriori informazioni, consulta i [modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector](#) nella Amazon Inspector User Guide.

- Per API i dettagli, consulta AWS CLI Command [ListAssessmentRuns](#) Reference.

## **list-assessment-targets**

Il seguente esempio di codice mostra come utilizzare `list-assessment-targets`.

## AWS CLI

Per elencare gli obiettivi di valutazione

Il `list-assessment-targets` comando seguente elenca tutti gli obiettivi di valutazione esistenti:

```
aws inspector list-assessment-targets
```

Output:

```
{
  "assessmentTargetArns": [
    "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq"
  ]
}
```

Per ulteriori informazioni, consulta gli obiettivi di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [ListAssessmentTargetsReference](#).

## `list-assessment-templates`

Il seguente esempio di codice mostra come utilizzare `list-assessment-templates`.

## AWS CLI

Per elencare i modelli di valutazione

Il `list-assessment-templates` comando seguente elenca tutti i modelli di valutazione esistenti:

```
aws inspector list-assessment-templates
```

Output:

```
{
  "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"
  ]
}
```

Per ulteriori informazioni, consulta i modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [ListAssessmentTemplates](#) Reference.

## list-coverage-statistics

Il seguente esempio di codice mostra come utilizzare `list-coverage-statistics`.

### AWS CLI

Esempio 1: per elencare le statistiche di copertura per gruppi

L'`list-coverage-statistics` esempio seguente elenca le statistiche di copertura del proprio AWS ambiente per gruppi.

```
aws inspector2 list-coverage-statistics \
  --group-by RESOURCE_TYPE
```

Output:

```
{
  "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
}

```

Per ulteriori informazioni, consulta la sezione [Valutazione della copertura di Amazon Inspector del AWS tuo ambiente](#) nella Amazon Inspector User Guide.

Esempio 2: per elencare le statistiche di copertura per tipo di risorsa

L'`list-coverage-statistics` seguente elenca le statistiche di copertura dell' AWS ambiente in uso per tipo di risorsa.

```

aws inspector2 list-coverage-statistics
  --filter-criteria '{"resourceType":
[{"comparison": "EQUALS", "value": "AWS_ECR_REPOSITORY"}]}'
  --group-by SCAN_STATUS_REASON

```

Output:

```

{
  "countsByGroup": [
    {
      "count": 27,
      "groupKey": "SUCCESSFUL"
    }
  ],
  "totalCounts": 27
}

```

Per ulteriori informazioni, consulta la sezione [Valutazione della copertura di Amazon Inspector del AWS tuo ambiente](#) nella Amazon Inspector User Guide.

Esempio 3: per elencare le statistiche di copertura in base al nome del repository ECR

L'`list-coverage-statistics` esempio seguente elenca le statistiche di copertura dell' AWS ambiente in base al nome del ECR repository.

```
aws inspector2 list-coverage-statistics
  --filter-criteria '{"ecrRepositoryName":
[{"comparison": "EQUALS", "value": "debian"}]}'
  --group-by SCAN_STATUS_REASON
```

Output:

```
{
  "countsByGroup": [
    {
      "count": 3,
      "groupKey": "SUCCESSFUL"
    }
  ],
  "totalCounts": 3
}
```

Per ulteriori informazioni, consulta la sezione [Valutazione della copertura di Amazon Inspector del AWS tuo](#) ambiente nella Amazon Inspector User Guide.

- Per API i dettagli, consulta Command [ListCoverageStatistics](#) Reference AWS CLI .

## list-coverage

Il seguente esempio di codice mostra come utilizzare `list-coverage`.

AWS CLI

Esempio 1: per elencare i dettagli di copertura relativi all'ambiente in uso

L'`list-coverage` esempio seguente elenca i dettagli della copertura dell'ambiente.

```
aws inspector2 list-coverage
```

Output:

```
{
  "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"
    }
  ]
}

```

Esempio 2: per elencare i dettagli della copertura sul tipo di risorsa della funzione Lambda

L'list-coverageesempio seguente elenca i dettagli del tipo di risorsa della funzione Lambda.

```

aws inspector2 list-coverage
  --filter-criteria '{"resourceType":
[{"comparison": "EQUALS", "value": "AWS_LAMBDA_FUNCTION"}]}'

```

Output:

```

{
  "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"
  }
]
}

```

- Per API i dettagli, vedere [ListCoverage](#) in AWS CLI Command Reference.

## list-delegated-admin-accounts

Il seguente esempio di codice mostra come utilizzare `list-delegated-admin-accounts`.

### AWS CLI

Per elencare informazioni sull'account amministratore delegato dell'organizzazione

L'`list-delegated-admin-accounts` esempio seguente elenca le informazioni sull'account amministratore delegato dell'organizzazione.

```
aws inspector2 list-delegated-admin-accounts
```

Output:

```

{
  "delegatedAdminAccounts": [
    {
      "accountId": "123456789012",
      "status": "ENABLED"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Designazione di un amministratore delegato per Amazon Inspector nella Amazon](#) Inspector User Guide.

- Per i API dettagli, consulta [ListDelegatedAdminAccounts](#) Command Reference.AWS CLI

## list-event-subscriptions

Il seguente esempio di codice mostra come utilizzare `list-event-subscriptions`.

### AWS CLI

Per elencare le sottoscrizioni agli eventi

Il `list-event-subscriptions` comando seguente elenca tutte le sottoscrizioni agli eventi per il modello di valutazione con: ARN `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0`

```
aws inspector list-event-subscriptions --resource-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0
```

Output:

```
{
  "subscriptions": [
    {
      "eventSubscriptions": [
        {
          "event": "ASSESSMENT_RUN_COMPLETED",
          "subscribedAt": 1459455440.867
        }
      ],
      "resourceArn": "arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-7sbz2Kz0",
      "topicArn": "arn:aws:sns:us-west-2:123456789012:exampletopic"
    }
  ]
}
```

Per ulteriori informazioni, consulta i modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [ListEventSubscriptions](#) Reference.

## list-filters

Il seguente esempio di codice mostra come utilizzare `list-filters`.

## AWS CLI

Per elencare i filtri associati all'account utilizzato per attivare Amazon Inspector

`list-filters` Gli esempi seguenti elencano i filtri associati all'account che hai usato per attivare Amazon Inspector.

```
aws inspector2 list-filters
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Filtrare i risultati di Amazon Inspector](#) nella Amazon Inspector User Guide.

- Per API i dettagli, consulta Command [ListFilters](#)Reference AWS CLI .

## list-findings

Il seguente esempio di codice mostra come utilizzare `list-findings`.

### AWS CLI

Per elencare i risultati

Il `list-findings` comando seguente elenca tutti i risultati generati:

```
aws inspector list-findings
```

Output:

```
{
  "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"
  ]
}
```

Per ulteriori informazioni, consulta Amazon Inspector Findings nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [ListFindings](#)Reference.



## list-rules-packages

Il seguente esempio di codice mostra come utilizzare `list-rules-packages`.

### AWS CLI

Per elencare i pacchetti di regole

Il `list-rules-packages` comando seguente elenca tutti i pacchetti di regole Inspector disponibili:

```
aws inspector list-rules-packages
```

Output:

```
{
  "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"
  ]
}
```

Per ulteriori informazioni, consulta Amazon Inspector Rules Packages and Rules nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [ListRulesPackages](#) Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag della risorsa

Il `list-tags-for-resource` comando seguente elenca tutti i tag associati al modello di valutazione con il comando ARN of `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
```

Output:

```
{
  "tags": [
    {
      "key": "Name",
      "value": "Example"
    }
  ]
}
```

Per ulteriori informazioni, consulta i modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [ListTagsForResource](#) Reference.

## list-usage-totals

Il seguente esempio di codice mostra come utilizzare `list-usage-totals`.

AWS CLI

Per elencare i totali di utilizzo negli ultimi 30 giorni

`list-usage-totals` Gli esempi seguenti elencano i totali di utilizzo negli ultimi 30 giorni.

```
aws inspector2 list-usage-totals
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Monitoraggio dell'utilizzo e dei costi in Amazon Inspector](#) nella Amazon Inspector User Guide.

- Per API i dettagli, consulta AWS CLI Command [ListUsageTotals](#)Reference.

## preview-agents

Il seguente esempio di codice mostra come utilizzare `preview-agents`.

### AWS CLI

Per visualizzare in anteprima gli agenti

Il `preview-agents` comando seguente visualizza in anteprima gli agenti installati sulle EC2 istanze che fanno parte dell'obiettivo di valutazione con: ARN `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
```

Output:

```
{
  "agentPreviews": [
    {
      "agentId": "i-49113b93"
    }
  ]
}
```

Per ulteriori informazioni, consulta gli obiettivi di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [PreviewAgentsReference](#).

## register-cross-account-access-role

Il seguente esempio di codice mostra come utilizzare `register-cross-account-access-role`.

### AWS CLI

Per registrare il ruolo di accesso multiaccount

Il `register-cross-account-access-role` comando seguente registra il IAM ruolo con `arn:aws:iam::123456789012:role/inspector` quello utilizzato da Amazon Inspector per

EC2 elencare le istanze all'inizio dell'esecuzione di valutazione o quando si chiama il comando `preview-agents`: ARN

```
aws inspector register-cross-account-access-role --role-arn arn:aws:iam::123456789012:role/inspector
```

Per ulteriori informazioni, consulta Configurazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [RegisterCrossAccountAccessRoleReference](#).

## remove-attributes-from-findings

Il seguente esempio di codice mostra come utilizzare `remove-attributes-from-findings`.

### AWS CLI

Per rimuovere gli attributi dai risultati

Il `remove-attributes-from-finding` comando seguente rimuove l'attributo con la chiave `Example` e il valore di `example` dal risultato con ARN di `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
```

Output:

```
{
  "failedItems": {}
}
```

Per ulteriori informazioni, consulta Amazon Inspector Findings nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [RemoveAttributesFromFindingsReference](#).

## set-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `set-tags-for-resource`.

## AWS CLI

Per impostare i tag per una risorsa

Il `set-tags-for-resource` comando seguente imposta il tag con la chiave `Example` e il valore `example` di nel modello di valutazione con ARN `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
```

Per ulteriori informazioni, consulta i modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [SetTagsForResource](#)Reference.

## start-assessment-run

Il seguente esempio di codice mostra come utilizzare `start-assessment-run`.

### AWS CLI

Per avviare un'esecuzione di valutazione

Il `start-assessment-run` comando seguente avvia l'esecuzione di valutazione denominata `examplerun` utilizzando il modello di valutazione con il comando ARN `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
```

Output:

```
{
  "assessmentRunArn": "arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-j0oroxyY"
}
```

Per ulteriori informazioni, consulta i modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [StartAssessmentRun](#) Reference.

## stop-assessment-run

Il seguente esempio di codice mostra come utilizzare `stop-assessment-run`.

### AWS CLI

Per interrompere un'esecuzione di valutazione

Il `stop-assessment-run` comando seguente interrompe l'esecuzione della valutazione con ARN di `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-j0oroxY`:

```
aws inspector stop-assessment-run --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-j0oroxY
```

Per ulteriori informazioni, consulta i modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [StopAssessmentRun](#) Reference.

## subscribe-to-event

Il seguente esempio di codice mostra come utilizzare `subscribe-to-event`.

### AWS CLI

Per iscriversi a un evento

L'esempio seguente abilita il processo di invio di SNS notifiche Amazon sull'`ASSESSMENT_RUN_COMPLETED` evento all'argomento con il comando ARN di `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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta i [modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector](#) nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [SubscribeToEvent](#)Reference.

## **unsubscribe-from-event**

Il seguente esempio di codice mostra come utilizzare `unsubscribe-from-event`.

### AWS CLI

Per annullare l'iscrizione a un evento

Il `unsubscribe-from-event` comando seguente disabilita il processo di invio di SNS notifiche Amazon sull'`ASSESSMENT_RUN_COMPLETED` evento all'argomento con il comando 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
```

Per ulteriori informazioni, consulta i modelli di valutazione e le esecuzioni di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [UnsubscribeFromEvent](#)Reference.

## **update-assessment-target**

Il seguente esempio di codice mostra come utilizzare `update-assessment-target`.

### AWS CLI

Per aggiornare un obiettivo di valutazione

Il `update-assessment-target` comando seguente aggiorna l'obiettivo di valutazione con l'ARN `arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX` e il nome `Example`, e il gruppo di risorse con il ARN `arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-yNbgL5Pt`:



```
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
```

Per ulteriori informazioni, consulta gli obiettivi di valutazione di Amazon Inspector nella guida Amazon Inspector.

- Per API i dettagli, consulta AWS CLI Command [UpdateAssessmentTarget](#) Reference.

## update-filter

Il seguente esempio di codice mostra come utilizzare `update-filter`.

### AWS CLI

Per aggiornare un filtro

L'`update-filter` esempio seguente aggiorna un filtro per omettere i risultati Lambda anziché i risultati ECR delle istanze.

```
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"
```

Output:

```
{  
  "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"
}
]
```

Per ulteriori informazioni, consulta la sezione [Gestione dei risultati in Amazon Inspector](#) nella Amazon Inspector User Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateFilter](#)Reference.

## AWS IoT esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS IoT.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **accept-certificate-transfer**

Il seguente esempio di codice mostra come utilizzare `accept-certificate-transfer`.

## AWS CLI

Accettare il certificato di un dispositivo trasferito da un altro AWS account

L'accept-certificate-transfer esempio seguente accetta il certificato di un dispositivo trasferito da un altro AWS account. Il certificato è identificato dal relativo ID.

```
aws iot accept-certificate-transfer \  
  --certificate-  
  id 488b6a7f2acdeb00a77384e63c4e40b18bEXAMPLEe57b7272ba44c45e3448142
```

Il comando non produce output.

Per ulteriori informazioni, consulta [Trasferire un certificato su un altro account](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [AcceptCertificateTransfer AWS CLI Command Reference](#).

## add-thing-to-billing-group

Il seguente esempio di codice mostra come utilizzare add-thing-to-billing-group.

### AWS CLI

Esempio 1: aggiungere un elemento per nome a un gruppo di fatturazione

L'add-thing-to-billing-group esempio seguente aggiunge l'oggetto denominato MyLightBulb al gruppo di fatturazione denominato GroupOne

```
aws iot add-thing-to-billing-group \  
  --billing-group-name GroupOne \  
  --thing-name MyLightBulb
```

Questo comando non produce alcun output.

Esempio 2: aggiungere un elemento by ARN a un gruppo di fatturazione

L'add-thing-to-billing-group esempio seguente aggiunge un oggetto con un valore specificato ARN a un gruppo di fatturazione con uno specificato. ARN Specificare un ARN è utile se lavori con più AWS regioni o account. Può aiutarti a garantire che tu stia effettuando l'aggiunta alla regione e all'account corretti.

```
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"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Billing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [AddThingToBillingGroup AWS CLI Command Reference](#).

## add-thing-to-thing-group

Il seguente esempio di codice mostra come utilizzare `add-thing-to-thing-group`.

### AWS CLI

Per aggiungere un elemento a un gruppo

L'`add-thing-to-thing-group` esempio seguente aggiunge l'oggetto specificato al gruppo di oggetti specificato.

```
aws iot add-thing-to-thing-group \  
  --thing-name MyLightBulb \  
  --thing-group-name LightBulbs
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [AddThingToThingGroup](#) in AWS CLI Command Reference.

## associate-targets-with-job

Il seguente esempio di codice mostra come utilizzare `associate-targets-with-job`.

### AWS CLI

Associare un gruppo di oggetti a un lavoro continuo

L'`associate-targets-with-job` esempio seguente associa il gruppo di oggetti specificato al lavoro continuo specificato.

```
aws iot associate-targets-with-job \  
  --targets "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \  
  --job-id "example-job-04"
```

Output:

```
{  
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",  
  "jobId": "example-job-04",  
  "description": "example continuous job"  
}
```

Per ulteriori informazioni, consulta [Creating and Managing Jobs \(CLI\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [AssociateTargetsWithJob AWS CLI Command Reference](#).

## attach-policy

Il seguente esempio di codice mostra come utilizzare `attach-policy`.

### AWS CLI

Esempio 1: Allegare una policy a un gruppo di oggetti

L'attach-policy esempio seguente associa la policy specificata a un gruppo di cose identificato dalla sua ARN.

```
aws iot attach-policy \  
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \  
  --policy-name "UpdateDeviceCertPolicy"
```

Il comando non produce output.

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developers Guide.

Esempio 2: per allegare una policy a un certificato

L'attach-policy esempio seguente associa la politica `UpdateDeviceCertPolicy` al principale specificato da un certificato.

```
aws iot attach-policy \  
  --policy-name UpdateDeviceCertPolicy \  
  --target "arn:aws:iot:us-west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"
```

Il comando non produce output.

Per ulteriori informazioni, consulta [Allegare una policy AWS IoT a un certificato di dispositivo](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [AttachPolicy AWS CLI Command Reference](#).

## attach-security-profile

Il seguente esempio di codice mostra come utilizzare `attach-security-profile`.

### AWS CLI

Per associare un profilo di sicurezza a tutti i dispositivi non registrati

L'esempio seguente associa il profilo di sicurezza AWS IoT Device Defender `Testprofile` denominato a tutti i dispositivi non registrati `us-west-2` nella regione di questo account. AWS

```
aws iot attach-security-profile \  
  --security-profile-name Testprofile \  
  --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [AttachSecurityProfile](#) in AWS CLI Command Reference.

## attach-thing-principal

Il seguente esempio di codice mostra come utilizzare `attach-thing-principal`.

### AWS CLI

Per allegare un certificato al tuo dispositivo

L'`attach-thing-principal` seguente allega un certificato all' `MyTemperatureSensor` oggetto. Il certificato è identificato da un ARN. Puoi trovare il certificato ARN per un certificato nella console AWS IoT.

```
aws iot attach-thing-principal \  
  --thing-name MyTemperatureSensor \  
  --principal arn:aws:iot:us-west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a35159767055278e8
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [How to Manage Things with the Registry](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [AttachThingPrincipal](#) in AWS CLI Command Reference.

## `cancel-audit-mitigation-actions-task`

Il seguente esempio di codice mostra come utilizzare `cancel-audit-mitigation-actions-task`.

AWS CLI

Per annullare un'attività relativa alle azioni di mitigazione degli audit

L'`cancel-audit-mitigation-actions-task` seguente annulla l'applicazione delle azioni di mitigazione per l'attività specificata. Non è possibile annullare attività già completate.

```
aws iot cancel-audit-mitigation-actions-task  
  --task-id "myActionsTaskId"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [CancelAuditMitigationActionsTask \(Mitigation Action Commands\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [CancelAuditMitigationActionsTask](#) in AWS CLI Command Reference.

## `cancel-audit-task`

Il seguente esempio di codice mostra come utilizzare `cancel-audit-task`.

## AWS CLI

Per annullare un'attività di controllo

L'`cancel-audit-task`esempio seguente annulla un'attività di controllo con l'ID dell'attività specificato. Non è possibile annullare un'attività completata.

```
aws iot cancel-audit-task \  
  --task-id a3aea009955e501a31b764abe1bebd3d
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Audit Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [CancelAuditTask](#) in AWS CLI Command Reference.

## `cancel-certificate-transfer`

Il seguente esempio di codice mostra come utilizzare `cancel-certificate-transfer`.

## AWS CLI

Per annullare il trasferimento di un certificato su un altro AWS account

L'`cancel-certificate-transfer`esempio seguente annulla il trasferimento del trasferimento del certificato specificato. Il certificato è identificato da un ID di certificato. Puoi trovare l'ID di un certificato nella console AWS IoT.

```
aws iot cancel-certificate-transfer \  
  --certificate-  
  id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78add5e605d630e05c7fc8
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Trasferire un certificato su un altro account](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [CancelCertificateTransfer AWS CLI](#) Command Reference.

## `cancel-job-execution`

Il seguente esempio di codice mostra come utilizzare `cancel-job-execution`.



## AWS CLI

Per annullare l'esecuzione di un lavoro su un dispositivo

L'`cancel-job-execution` esempio seguente annulla l'esecuzione del processo specificato su un dispositivo. Se il lavoro non è nello QUEUED stato, è necessario aggiungere il `--force` parametro.

```
aws iot cancel-job-execution \  
  --job-id "example-job-03" \  
  --thing-name "MyRPi"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creating and Managing Jobs \(CLI\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [CancelJobExecution AWS CLI Command Reference](#).

## cancel-job

Il seguente esempio di codice mostra come utilizzare `cancel-job`.

## AWS CLI

Per annullare un lavoro

L'`cancel-job` esempio seguente annulla il lavoro specificato.

```
aws iot cancel-job \  
  --job-job "example-job-03"
```

Output:

```
{  
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-03",  
  "jobId": "example-job-03",  
  "description": "example job test"  
}
```

Per ulteriori informazioni, consulta [Creating and Managing Jobs \(CLI\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [CancelJob AWS CLI Command Reference](#).

## clear-default-authorizer

Il seguente esempio di codice mostra come utilizzare `clear-default-authorizer`.

### AWS CLI

Per cancellare l'autorizzatore predefinito

L'`clear-default-authorizer` esempio seguente cancella l'autorizzatore personalizzato predefinito attualmente configurato. Dopo aver eseguito questo comando, non esiste un autorizzatore predefinito. Quando si utilizza un autorizzatore personalizzato, è necessario specificarlo per nome nelle intestazioni della HTTP richiesta.

```
aws iot clear-default-authorizer
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [ClearDefaultAuthorizer](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [ClearDefaultAuthorizer](#) in AWS CLI Command Reference.

## confirm-topic-rule-destination

Il seguente esempio di codice mostra come utilizzare `confirm-topic-rule-destination`.

### AWS CLI

Per confermare la destinazione di una regola tematica

L'`confirm-topic-rule-destination` esempio seguente conferma la destinazione di una regola tematica con un token di conferma ricevuto presso un HTTP endpoint.

```
aws iot confirm-topic-rule-destination \
  --confirmation-token "AYADeIcmtq-
ZkxfpiWIQqHWM5ucAXwABABVhd3MtY3J5cHRvLXB1YmxpYy1rZXkAREFxY1E0Um1GeDg0V21BZWZ1VjZtZWFRVUJJUkt
aywpPqg8YEsa1LD4B40aJ2s1wEHKMybiF1Ro0ZzYisI0IvsLzQY5UmCkq3tV-3f7-
nKfosgIAAAAADAAAEEAAAAAAAAAAAAAAAAAAAAi9RMgy-
V19V9m6Iw2xfbw_____wAAAAEAAAAAAAAAAAAAAAAAAEAAAB1hw4SokgUcxiJ3gT06n50NLJVpzyQR1UmPIj5sShqXEQGcC
iufgrzTeP18RZY0Wr006Aj9DiVzJZx-1iD6Pu-
```

```
G6PUw1ka07Knzs2B4AD0qfrHUF4pYRTvyUgBnMGUCMQC8ZRmhKqntd_c6Kgrow3bMUDBvNqo2qZr8Z8Jm2rzgseR01AnPIetJ803Z4I1I1F8xXlcdPGP-PV1d0XFemyL8g"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Confermare la destinazione di una regola tematica](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [ConfirmTopicRuleDestination AWS CLI Command Reference](#).

## create-audit-suppression

Il seguente esempio di codice mostra come utilizzare `create-audit-suppression`.

### AWS CLI

Per creare una soppressione dei risultati di controllo

L'esempio seguente crea una soppressione dei risultati di controllo per una politica denominata "virtualMachinePolicy" che è stata contrassegnata come eccessivamente permissiva.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Audit finding suppressions](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateAuditSuppression](#) Reference.

## create-authorizer

Il seguente esempio di codice mostra come utilizzare `create-authorizer`.

### AWS CLI

Per creare un autorizzatore personalizzato

L'`create-authorizer` esempio seguente crea un autorizzatore personalizzato che utilizza la funzione Lambda specificata come parte di un servizio di autenticazione personalizzato.

```
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-----
MIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEA1uJOB4lQPgG/lM6ZfIwo
Z+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0U8Eq9xb503kRSmyIwTzwzm/f4Gf0Y
ZUloJ+t3PUUwHrmbYTAgrCUgRFygjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
zw0BKPeic0asNJpqT8PkBbRaKylEJh5oo81NDHHmVtbBm5A5YiJjqYXLaVAowKzZ
+GqsNvAQ9Jy1wI2VrEa10fL8f1DB/BJLm7zjpfPOHDJQgID0XnZwAlNnZcOhCwIx
50g2LW20y9R/dmqtDmJiVP97Z4GykxPvwLYHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1
lQIDAQAB
-----END PUBLIC KEY-----"
```

Output:

```
{
  "authorizerName": "CustomAuthorizer",
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer2"
}
```

Per ulteriori informazioni, [CreateAuthorizer](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [CreateAuthorizer](#) in AWS CLI Command Reference.

## **create-billing-group**

Il seguente esempio di codice mostra come utilizzare `create-billing-group`.

### AWS CLI

Per creare un gruppo di fatturazione

L'`create-billing-group` esempio seguente crea un gruppo di fatturazione semplice denominato `GroupOne`

```
aws iot create-billing-group \  
  --billing-group-name GroupOne
```

Output:

```
{  
  "billingGroupName": "GroupOne",  
  "billingGroupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne",  
  "billingGroupId": "103de383-114b-4f51-8266-18f209ef5562"  
}
```

Per ulteriori informazioni, consulta [Billing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [CreateBillingGroup AWS CLI Command Reference](#).

## create-certificate-from-csr

Il seguente esempio di codice mostra come utilizzare `create-certificate-from-csr`.

AWS CLI

Per creare un certificato del dispositivo da una richiesta di firma del certificato (CSR)

L'`create-certificate-from-csr` seguente crea un certificato del dispositivo da un CSR. È possibile utilizzare il `openssl` comando per creare un CSR.

```
aws iot create-certificate-from-csr \  
  --certificate-signing-request=file://certificate.csr
```

Output:

```
{  
  "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/  
c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbbee1428d216d54d53ac9",  
  "certificateId":  
  "c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbbee1428d216d54d53ac9",  
  "certificatePem": "<certificate-text>"  
}
```

Per ulteriori informazioni, [CreateCertificateFromCSR](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [CreateCertificateFromCsr](#) in AWS CLI Command Reference.

## create-custom-metric

Il seguente esempio di codice mostra come utilizzare `create-custom-metric`.

### AWS CLI

Per creare una metrica personalizzata pubblicata dai tuoi dispositivi su Device Defender

L'esempio seguente crea una metrica personalizzata che misura la percentuale della batteria.

```
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"
```

Output:

```
{
  "metricName": "batteryPercentage",
  "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/
batteryPercentage"
}
```

Per ulteriori informazioni, consulta [Metriche personalizzate](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [CreateCustomMetric AWS CLI Command Reference](#).

## create-dimension

Il seguente esempio di codice mostra come utilizzare `create-dimension`.

### AWS CLI

Per creare una dimensione

Quanto segue `create-dimension` crea una dimensione con un singolo filtro per argomento chiamato `TopicFilterForAuthMessages`.

```
aws iot create-dimension \  
  --name TopicFilterForAuthMessages \  
  --type TOPIC_FILTER \  
  --string-values device/+/auth
```

Output:

```
{  
  "name": "TopicFilterForAuthMessages",  
  "arn": "arn:aws:iot:eu-west-2:123456789012:dimension/TopicFilterForAuthMessages"  
}
```

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [CreateDimension](#) in AWS CLI Command Reference.

## create-domain-configuration

Il seguente esempio di codice mostra come utilizzare `create-domain-configuration`.

AWS CLI

Per creare una configurazione di dominio

L'esempio seguente crea una configurazione AWS di dominio gestito con un tipo di servizio di DATA.

```
aws iot create-domain-configuration \  
  --domain-configuration-name "additionalDataDomain" \  
  --service-type "DATA"
```

Output:

```
{  
  "domainConfigurationName": "additionalDataDomain",  
  "domainConfigurationArn": "arn:aws:iot:us-  
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"  
}
```

Per ulteriori informazioni, consulta [Configurable Endpoints](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateDomainConfiguration](#) Reference.

## create-dynamic-thing-group

Il seguente esempio di codice mostra come utilizzare `create-dynamic-thing-group`.

### AWS CLI

Per creare un gruppo di oggetti dinamico

L'esempio seguente crea un gruppo di oggetti dinamico che contiene qualsiasi cosa con un attributo di temperatura maggiore di 60 gradi. È necessario abilitare l'indicizzazione della flotta AWS IoT prima di poter utilizzare i gruppi di oggetti dinamici.

```
aws iot create-dynamic-thing-group \  
  --thing-group-name "RoomTooWarm" \  
  --query-string "attributes.temperature>60"
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Dynamic Thing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [CreateDynamicThingGroup](#) in AWS CLI Command Reference.

## create-job

Il seguente esempio di codice mostra come utilizzare `create-job`.

### AWS CLI

Esempio 1: creare un lavoro



L'create-jobesempio seguente crea un semplice processo AWS IoT che invia un JSON documento al MyRaspberryPi dispositivo.

```
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
```

Output:

```
{  
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",  
  "jobId": "example-job-01",  
  "description": "example job test"  
}
```

Esempio 2: creare un lavoro continuo

L'create-jobesempio seguente crea un processo che continua a essere eseguito dopo che gli elementi specificati come obiettivi hanno completato il lavoro. In questo esempio, la destinazione è un gruppo di oggetti, quindi quando vengono aggiunti nuovi dispositivi al gruppo, il processo continuo viene eseguito su tali nuovi elementi.

```
aws iot create-job --job-id «example-job-04» --targets «arn:aws:iot:us-west- 2:123456789012: -  
job.json --description «esempio di lavoro continuo» --target-selection thinggroup/DeadBulbs" --  
document file://example CONTINUOUS
```

Output:

```
{  
  "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",  
  "jobId": "example-job-04",  
  "description": "example continuous job"  
}
```

Per ulteriori informazioni, consulta [Creating and Managing Jobs \(CLI\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [CreateJob AWS CLI Command Reference](#).

## create-keys-and-certificate

Il seguente esempio di codice mostra come utilizzare `create-keys-and-certificate`.

### AWS CLI

Per creare una RSA key pair ed emettere un certificato X.509

Quanto segue `create-keys-and-certificate` crea una coppia di RSA chiavi a 2048 bit ed emette un certificato X.509 utilizzando la chiave pubblica emessa. Poiché è l'unica volta in cui l'AWS IoT fornisce la chiave privata per questo certificato, assicurati di conservarlo in un luogo sicuro.

```
aws iot create-keys-and-certificate \
  --certificate-pem-outfile "myTest.cert.pem" \
  --public-key-outfile "myTest.public.key" \
  --private-key-outfile "myTest.private.key"
```

Output:

```
{
  "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
  "certificateId":
  "9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
  "certificatePem": "
-----BEGIN CERTIFICATE-----
MIICiTCCEXAMPLE6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAGYEXAMPLEAwDgYDVoQHEwDQZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDA5BgNVBAsTC01BTSEXAMPLE2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAd
BgkqhkiG9w0BCQEWEG5vb251QGFTYEXAMPLEb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCEXAMPLEJBgNVBAGTAldBMRAwDgYD
VQQHEwDQZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDAEXAMPLEsTC01BTSBDb25z
b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAdBgkqhkiG9w0BCQEXAMPLE251QGFT
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+aEXAMPLE
EXAMPLEfEvySwTc2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZEXAMPLELg5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZncvQAEXAMPLEWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUHVvXyUntneD9+h8Mg9qEXAMPLEEyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDEXAMPLEBHjJnyp3780D8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n",
  "keyPair": {
```

```

    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiEXAMPEQEFAA0CAQ8AMIIBCgKCAQEAEEXAMPLE1nnyJwKSMHw4h\nMMEXAMPLEEuN/
dMAS3fyce8DW/4+EXAMPLEYjmoF/YVF/gHr99VEEXAMPLE5VF13\n59VK7cEXAMPLE67GK+y+jikqX0gHh/
xJTwo
+sGpWEXAMPLEDz18x0d2ka4tCzuWEXAMPLEEahJbYkCPUBSU8opVkr7qkEXAMPLE1DR6sx2Hocli00Ltu6Fkw91swQWEX
\GB3ZPrNh0PzQYvjUStZeccyNCx2EXAMPLEVp9mQ0UXP6p1fgxwKRX2fEXAMPLEDa
\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"
  }
}

```

Per ulteriori informazioni, consulta [Creare e registrare un certificato per dispositivi AWS IoT](#) nella Guida per sviluppatori AWS IoT.

- Per API i dettagli, consulta Command [CreateKeysAndCertificate](#) Reference AWS CLI .

## create-mitigation-action

Il seguente esempio di codice mostra come utilizzare `create-mitigation-action`.

### AWS CLI

Per creare un'azione di mitigazione

L'`create-mitigation-action` seguente definisce un'azione di mitigazione denominata `AddThingsToQuarantineGroup1Action` che, quando applicata, sposta gli elementi nel gruppo di cose denominato `QuarantineGroup1`. Questa azione ha la precedenza sui gruppi di oggetti dinamici.

```
aws iot create-mitigation-action --cli-input-json file::params.json
```

Contenuto di `params.json`.

```

{
  "actionName": "AddThingsToQuarantineGroup1Action",
  "actionParams": {
    "addThingsToThingGroupParams": {
      "thingGroupNames": [
        "QuarantineGroup1"
      ],
    },
  },
}

```

```

        "overrideDynamicGroups": true
      }
    },
    "roleArn": "arn:aws:iam::123456789012:role/service-role/
MoveThingsToQuarantineGroupRole"
  }

```

### Output:

```

{
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroup1Action",
  "actionId": "992e9a63-a899-439a-aa50-4e20c52367e1"
}

```

Per ulteriori informazioni, consulta [CreateMitigationAction \(Mitigation Action Commands\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [CreateMitigationAction](#) in AWS CLI Command Reference.

## create-ota-update

Il seguente esempio di codice mostra come utilizzare `create-ota-update`.

### AWS CLI

Per creare un OTA aggiornamento da utilizzare con Amazon Free RTOS

L'esempio seguente crea un AWS IoT OTAUpdate su un gruppo target di cose o gruppi. Questo fa parte di un RTOS over-the-air aggiornamento Amazon Free che consente di distribuire nuove immagini del firmware su un singolo dispositivo o un gruppo di dispositivi.

```

aws iot create-ota-update \
  --cli-input-json file://create-ota-update.json

```

Contenuto di `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"
}
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, vedere [C reateOTAUpdate](#) in AWS IoT API Reference.

- Per API i dettagli, vedere [CreateOtaUpdate](#) in AWS CLI Command Reference.

## create-policy-version

Il seguente esempio di codice mostra come utilizzare `create-policy-version`.

## AWS CLI

Per aggiornare una politica con una nuova versione

L'create-policy-versionesempio seguente aggiorna una definizione di policy, creando una nuova versione della policy. Questo esempio imposta inoltre la nuova versione come predefinita.

```
aws iot create-policy-version \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-document file://policy.json \  
  --set-as-default
```

Contenuto di policy.json.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "iot:UpdateCertificate",  
      "Resource": "*"  
    }  
  ]  
}
```

Output:

```
{  
  "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  
}
```

Per ulteriori informazioni, consulta [AWS IoT Policies](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [CreatePolicyVersion](#) in AWS CLI Command Reference.

## create-policy

Il seguente esempio di codice mostra come utilizzare create-policy.

## AWS CLI

Per creare una policy AWS IoT

L'create-policyesempio seguente crea una policy AWS IoT denominata TemperatureSensorPolicy. Il policy.json file contiene istruzioni che consentono azioni politiche AWS IoT.

```
aws iot create-policy \  
  --policy-name TemperatureSensorPolicy \  
  --policy-document file://policy.json
```

Contenuto di 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"
        ]
    }
]
}

```

## Output:

```

{
  "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\": [

```





```

T96cRBSWnWmon0WdY0GKVzni0CA\n+iyGudgrFKm7Eae/
v18oXrf82Kt0AG04xG0KE2WKYHsT1fx3c9xZhLXP/eX
Lhv00\n+1Gp0WVw9PbhKfrxliKJ5q6sL5nVUaUHq6h1QPYwsATe0vAp3u0ak5zgTyL0fg7Y
\nPyKk6VYwLW62r+v
YBSForEM0Ahkq3LsP/rjxpeKmi2W41PVS6oFZRKcD+H1Kyil5\nAgMBAAGjIDAeMAwGA1UdEwEB/
wQCMAAwDgYDV
R0PAQH/BAQDAgeAMA0GCSqGSIb3\nDQEBcWUAA4IBAQAQgix2k6nVqbZFKq97/fZBzLGS0dyz5rT/
E41cDIRX+1j
EPW41\nnw0D+2sXheCZLZZnSkvIiP74IToNeXDrjdcaodeGFVHIElRjhMIq+4ZebPbRLtidF
\nRc2hfcTAlqq9Z6v
5Vk6BeM1tu0RqH1wPoVUccLPya8EjNCbnJZUmGd0frN/Y9pho\n5ikV+HPeZhG/k6dhE2GsQJyKfVHL/
uBgKSily
1bRyWU1r6qcpWBNBHjUoD7Hg0wD
\nnzMh4XRb2FQDsqFalkCSYmeL8IVC49sgPD90typ5uteGMTy62usAAUQdq/f
ZvrWg\n0kFpwMVnGKVKT7Kg0kK0LzKW0BB2Jm4/gmrJ\n-----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCg
KCAQEAwYSiPeJLSi6k8J4/msjq
\nUwCbfzer0iCQ2b5a2I5AtB08M2nmN06a1pNN0tvb1M1bhDlx10F2W4oYKYN
pun8\n2pFpvf8KY8xPZ8ufsZDx1R+Fp8M+8iuZvEtGoC0/enEQUl1pqJzlnWNBilc54tA
\nngPoshrnYKxSpuxGn
v79fKF63/NirTgBjuMRtChNlimEXAMPLE3PcWYZVz/3ly4b9\nNPPRqdFlcPT24Sn68ZYiieaurC
+Z1VG1B6uoZU
D2MLAE3jrwKd7tGp0c4E8i9H40\n2D8ip0lWMC1utq/
lWAUhaKxDDgIZKty7D/648aRCpotluJT1UuqBWUSnA/h9
Ssop\nEQIDAQAB\n-----END PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQEAwYSiPeJLSi6k8J4/
msjqtUwCbfzer0iCQ2b5a2I5AtB08M2n
\nmN06a1pNN0tvb1M1bhDlx10F2W4oYKYNpun82pFpvf8KY8xPZ8ufsZ
Dx1R+Fp8M+\n8iuZvEtGoC0/enEQUl1pqJzlnWNBilc54tAgPoshrnYKxSpuxGnv79fKF63/Nir
\nTgBjuMRtCh
NlimB7E9X8d3PcWYZVz/3ly4b9NPPRqdFlcPT24Sn68ZYiieaurC+Z
\n1VG1B6uoZUD2MLAE3jrwKd7tGp0c4E8i
9H402D8ip0lWMC1utq/lWAUhaKxDDgIZ\nKty7D/648aRCpotluJT1UuqBWUSnA/
h9SsopeQIDAQABAoIBAEAybn
QUtx9T2/nK\nntZT2pA4iugecxI4dz+DmT0XVxs5VJmrx/
nBSq6ejXExEpSIM04RY7LE3ZdJcnd56\nF7tQkkY7yR
VzfxHeXFU1kr0IPuxWebN0rRoPZr+1RSer+ww2aBC525+88pVuR6tM
\nm3pgkrR2ycCj9Fd0UoQxdjHBHaM5PDmJ
9aSxCKdg3nReepeGwsR2TQA+m2vVxWk7\nnou0+91eTOP+/QfP7P8Zj0Ik02XivlRcVDyN/
E4QXPkuIkM/8vS8VK+

```

```

E9pATQ0MtB\n21w8R/YU5AJd6j1EXAMPLEGU2UzRzInNWILtkPPPqgqXXhx0f+mxByjcMa1VJk0L
\nh0G2R0UCgY
EA+R0cHNHy/XbsP7Fih0hEh+6Q2QxQ2ncBUPYbBazrR8Hn+7SCICQK
\nVyYfd8Ajfq3e7RsKVL5S1MBp7S1idxak
bIn28fKfPn62DaemGCIOyDgLPf+eUxBx
\nngzbCiBZga8brfurza43UZjKZLpg3hq721+FeAiXi1Nma4Yr9YWEHEN
8CgYEAXuWt\npzdWwmsiFzfsAw0sy9ySDA/xr5WRWzJyAqUsjsks6rxNzWebpufnYHcmtW7pLdqM
\nkboHwN2pXa
kmZvrk2nKkEMq5brBYGDxuxDe+V369Bianx8aZFyIsckA70wXW1w1h
\nngRC5rQ4X0gp3+Jmw7eA08LRYDjaN846+
Qbt02KcCgYAWS0UL51bijQR0ZwI0dz27\nFQVuCAYsp748aurcRTACCj8jbnK/
QbqTNlxWsaH7ssBjZKo2D5sAqY
BRtASW0Dab\naHXsDhVm2Jye+ESLoHMaCLoyCkT3118yqXIcEDStM07f01Ryag164EiJvSIrMfny\nNL/
fXVjCSH
/udCxdzPt+7QKBgQC+LAD7rxdr4J9538hTqpc4XK9vxRbrMXEH55XH
\nHbMa2x0NZXpmeTgEQBukyohCVceyRhK9
i0e6irZTjVXgh0eoTpC8VXkzcnzouTiQ
\nfQQSGfnp7Ioe6UIz23715pKduszSNkMSKrG924ktv7CyDBF1gBQI5g
aDoHnddJBJ\nnPRTIZQKBgA8MASXtTxQntRwXXzR92U0vAighiuRkB/mx9jQpUcK1qiqHbkAMqgNF
\nPFCBYIUbFT
iYKKKeJNbyJQvjfsJcKAnaFJ+RnTxk0Q6Wjm20peJ/ii4QiDdnigoE\nnvd1c5cFQewWb4/
zqAtPdinkPLN94ileI
79XQdc7R1J0jpgTimL+V\n-----END RSA PRIVATE KEY-----\n"
  },
  "expiration": 1595955066.0
}

```

Per ulteriori informazioni, consulta la sezione [Provisioning by trusted user](#) nella AWS IoT Core Developers Guide.

- Per API i dettagli, vedere [CreateProvisioningClaim](#) in AWS CLI Command Reference.

## create-provisioning-template-version

Il seguente esempio di codice mostra come utilizzare `create-provisioning-template-version`.

### AWS CLI

Per creare una versione del modello di provisioning

L'esempio seguente crea una versione per il modello di provisioning specificato. Il corpo della nuova versione viene fornito nel file `template.json`.

```
aws iot create-provisioning-template-version \  
  --template-name widget-template \  
  --template-body file://template.json
```

Contenuto di `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"]}
    }
}
}
}

```

### Output:

```

{
    "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-
template",
    "templateName": "widget-template",
    "versionId": 2,
    "isDefaultVersion": false
}

```

Per ulteriori informazioni, consulta [AWS IoT Secure Tunneling](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateProvisioningTemplateVersion](#)Reference.

## create-provisioning-template

Il seguente esempio di codice mostra come utilizzare `create-provisioning-template`.

### AWS CLI

Per creare un modello di provisioning

L'esempio seguente crea un modello di provisioning come definito dal file `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
```

Contenuto di `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"]}
    }
}
}

```

Output:

```
{
  "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-
template",
  "templateName": "widget-template",
  "defaultVersionId": 1
}
```

Per ulteriori informazioni, consulta [AWS IoT Secure Tunneling](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateProvisioningTemplate](#) Reference.

## create-role-alias

Il seguente esempio di codice mostra come utilizzare `create-role-alias`.

### AWS CLI

Per creare un alias di ruolo

L'esempio seguente crea un alias di ruolo chiamato `LightBulbRole` per il ruolo specificato.

```
aws iot create-role-alias \
  --role-alias LightBulbRole \
  --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

Output:

```
{
  "roleAlias": "LightBulbRole",
  "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"
}
```

Per ulteriori informazioni, [CreateRoleAlias](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [CreateRoleAlias](#) in AWS CLI Command Reference.

## create-scheduled-audit

Il seguente esempio di codice mostra come utilizzare `create-scheduled-audit`.



## AWS CLI

Per creare un audit pianificato

L'`create-scheduled-audit` seguente crea un controllo pianificato che viene eseguito settimanalmente, il mercoledì, per verificare se i certificati CA o i certificati dei dispositivi sono in scadenza.

```
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
```

Output:

```
{  
  "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/  
WednesdayCertCheck"  
}
```

Per ulteriori informazioni, consulta [Audit Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [CreateScheduledAudit](#) in AWS CLI Command Reference.

## `create-security-profile`

Il seguente esempio di codice mostra come utilizzare `create-security-profile`.

### AWS CLI

Per creare un profilo di sicurezza

L'`create-security-profile` seguente crea un profilo di sicurezza che verifica se la larghezza di banda cellulare supera una soglia o se si verificano più di 10 errori di autorizzazione in un periodo di cinque minuti.

```
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}]]"
```

Output:

```
{
  "securityProfileName": "PossibleIssue",
  "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
PossibleIssue"
}
```

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [CreateSecurityProfile](#) in AWS CLI Command Reference.

## create-stream

Il seguente esempio di codice mostra come utilizzare `create-stream`.

### AWS CLI

Per creare uno stream per la distribuzione di uno o più file di grandi dimensioni in blocchi MQTT

L'`create-stream` esempio seguente crea uno stream per la distribuzione di uno o più file di grandi dimensioni suddivisi in blocchi. MQTT Uno stream trasporta i byte di dati in blocchi o blocchi impacchettati come messaggi da una fonte come MQTT S3. A un flusso possono essere associati uno o più file.

```
aws iot create-stream \
  --cli-input-json file://create-stream.json
```

Contenuto di `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"
}

```

Output:

```

{
    "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"
}

```

Per ulteriori informazioni, [CreateStream](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [CreateStream](#) in AWS CLI Command Reference.

## create-thing-group

Il seguente esempio di codice mostra come utilizzare `create-thing-group`.

### AWS CLI

Esempio 1: creare un gruppo di oggetti

L'`create-thing-group` esempio seguente crea un gruppo di oggetti denominato `LightBulbs` con una descrizione e due attributi.

```

aws iot create-thing-group \
  --thing-group-name LightBulbs \
  --thing-group-properties "thingGroupDescription=\"Generic bulb group\",
  attributePayload={attributes={Manufacturer=AnyCompany,wattage=60}}"

```

Output:

```

{

```

```
"thingGroupName": "LightBulbs",
"thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs",
"thingGroupId": "9198bf9f-1e76-4a88-8e8c-e7140142c331"
}
```

Esempio 2: creare un gruppo di cose che fa parte di un gruppo principale

Quanto segue `create-thing-group` crea un gruppo di oggetti denominato `HalogenBulbs` con un gruppo di oggetti principale denominato `LightBulbs`.

```
aws iot create-thing-group \
  --thing-group-name HalogenBulbs \
  --parent-group-name LightBulbs
```

Output:

```
{
  "thingGroupName": "HalogenBulbs",
  "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs",
  "thingGroupId": "f4ec6b84-b42b-499d-9ce1-4dbd4d4f6f6e"
}
```

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [CreateThingGroup](#) in AWS CLI Command Reference.

## create-thing-type

Il seguente esempio di codice mostra come utilizzare `create-thing-type`.

AWS CLI

Per definire un tipo di cosa

L'`create-thing-type` esempio seguente definisce un tipo di oggetto e gli attributi associati.

```
aws iot create-thing-type \
  --thing-type-name "LightBulb" \
  --thing-type-properties "thingTypeDescription=light bulb type,
searchableAttributes=wattage,model"
```

Output:

```
{
  "thingTypeName": "LightBulb",
  "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",
  "thingTypeId": "ce3573b0-0a3c-45a7-ac93-4e0ce14cd190"
}
```

Per ulteriori informazioni, consulta [Thing Types](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [CreateThingType](#) in AWS CLI Command Reference.

## create-thing

Il seguente esempio di codice mostra come utilizzare `create-thing`.

### AWS CLI

Esempio 1: creare un record di oggetti nel registro

L'`create-things`esempio seguente crea una voce per un dispositivo nel registro degli AWS oggetti IoT.

```
aws iot create-thing \
  --thing-name SampleIoTThing
```

Output:

```
{
  "thingName": "SampleIoTThing",
  "thingArn": "arn:aws:iot:us-west-2: 123456789012:thing/SampleIoTThing",
  "thingId": " EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE "
}
```

Esempio 2: definire un oggetto associato a un tipo di oggetto

L'`create-things`esempio seguente crea un oggetto con il tipo di oggetto specificato e i relativi attributi.

```
aws iot create-thing \
  --thing-name "MyLightBulb" \
  --thing-type-name "LightBulb" \
  --attribute-payload '{"attributes": {"wattage": "75", "model": "123"}}'
```

**Output:**

```
{
  "thingName": "MyLightBulb",
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
  "thingId": "40da2e73-c6af-406e-b415-15acae538797"
}
```

Per ulteriori informazioni, consulta [How to Manage Things with the Registry](#) and [Thing Types](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [CreateThing](#) in AWS CLI Command Reference.

**create-topic-rule-destination**

Il seguente esempio di codice mostra come utilizzare `create-topic-rule-destination`.

**AWS CLI**

Per creare una destinazione per le regole dell'argomento

L'`create-topic-rule-destination` seguente crea una destinazione per le regole tematiche per un HTTP endpoint.

```
aws iot create-topic-rule-destination \
  --destination-configuration httpUrlConfiguration={confirmationUrl=https://
  example.com}
```

**Output:**

```
{
  "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"
    }
  }
}
```

```
}
```

Per ulteriori informazioni, consulta [Creazione di una destinazione per le regole tematiche](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [CreateTopicRuleDestination AWS CLI Command Reference](#).

## create-topic-rule

Il seguente esempio di codice mostra come utilizzare `create-topic-rule`.

### AWS CLI

Per creare una regola che invii un SNS avviso Amazon

L'esempio seguente crea una regola che invia un SNS messaggio Amazon quando i valori del livello di umidità del suolo, rilevati nell'ombra di un dispositivo, sono bassi.

```
aws iot create-topic-rule \  
  --rule-name "LowMoistureRule" \  
  --topic-rule-payload file://plant-rule.json
```

L'esempio richiede il salvataggio JSON del codice seguente in un file denominato `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"  
    }  
  }  
}]  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creating an AWS IoT Rule](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [CreateTopicRule AWS CLI Command Reference](#).

## **delete-account-audit-configuration**

Il seguente esempio di codice mostra come utilizzare `delete-account-audit-configuration`.

### AWS CLI

Per disabilitare tutti i controlli di controllo per il tuo AWS account

L'`delete-account-audit-configuration` seguente ripristina le impostazioni predefinite per AWS IoT Device Defender per questo account, disabilitando tutti i controlli di controllo e cancellando i dati di configurazione. Elimina inoltre tutti gli audit pianificati per questo account. Usa questo comando con cautela.

```
aws iot delete-account-audit-configuration \  
  --delete-scheduled-audits
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Audit Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DeleteAccountAuditConfiguration](#) in AWS CLI Command Reference.

## **delete-audit-suppression**

Il seguente esempio di codice mostra come utilizzare `delete-audit-suppression`.

### AWS CLI

Per eliminare un audit, la soppressione dei risultati

L'`delete-audit-suppression` seguente elimina una soppressione dei risultati di controllo per DEVICE \_\_ \_CERTIFICATE. EXPIRING CHECK

```
aws iot delete-audit-suppression \  
  --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \  
  --device-name my-device
```



```
--resource-identifier deviceCertificateId="c7691e<shortened>"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Audit finding suppressions](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteAuditSuppression](#) Reference.

## delete-authorizer

Il seguente esempio di codice mostra come utilizzare `delete-authorizer`.

### AWS CLI

Per eliminare un autorizzatore personalizzato

L'`delete-authorizer` esempio seguente elimina l'autorizzatore denominato.

`CustomAuthorizer` Un autorizzatore personalizzato deve trovarsi `INACTIVE` nello stato in cui si trova prima di poterlo eliminare.

```
aws iot delete-authorizer \  
  --authorizer-name CustomAuthorizer
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [DeleteAuthorizer](#) la AWS IoT Developer Guide.

- Per API i dettagli, vedere [DeleteAuthorizer](#) in AWS CLI Command Reference.

## delete-billing-group

Il seguente esempio di codice mostra come utilizzare `delete-billing-group`.

### AWS CLI

Per eliminare un gruppo di fatturazione

L'`delete-billing-group` esempio seguente elimina il gruppo di fatturazione specificato. Puoi eliminare un gruppo di fatturazione anche se contiene uno o più elementi.

```
aws iot delete-billing-group \  
  --billing-group-name BillingGroupTwo
```

Il comando non produce output.

Per ulteriori informazioni, consulta [Billing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [DeleteBillingGroup AWS CLI Command Reference](#).

## **delete-ca-certificate**

Il seguente esempio di codice mostra come utilizzare `delete-ca-certificate`.

### AWS CLI

Per eliminare un certificato CA

L'`delete-ca-certificate` esempio seguente elimina il certificato CA con l'ID di certificato specificato.

```
aws iot delete-ca-certificate \  
  --certificate-  
  id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [DeleteCACertificate](#) in AWS IoT API Reference.

- Per API i dettagli, vedere [DeleteCaCertificate](#) in AWS CLI Command Reference.

## **delete-certificate**

Il seguente esempio di codice mostra come utilizzare `delete-certificate`.

### AWS CLI

Per eliminare il certificato di un dispositivo

L'`delete-certificate` esempio seguente elimina il certificato del dispositivo con l'ID specificato.

```
aws iot delete-certificate \  
  --certificate-  
  id c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddb3e1428d216d54d53ac9
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [DeleteCertificate](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [DeleteCertificate](#) in AWS CLI Command Reference.

## **delete-custom-metric**

Il seguente esempio di codice mostra come utilizzare `delete-custom-metric`.

### AWS CLI

Per eliminare una metrica personalizzata

L'`delete-custom-metric` esempio seguente elimina una metrica personalizzata.

```
aws iot delete-custom-metric \  
  --metric-name batteryPercentage \  
  --region us-east-1
```

Output:

```
HTTP 200
```

Per ulteriori informazioni, consulta [Metriche personalizzate](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [DeleteCustomMetric AWS CLI](#) Command Reference.

## **delete-dimension**

Il seguente esempio di codice mostra come utilizzare `delete-dimension`.

### AWS CLI

Per eliminare una dimensione

L'`delete-dimension` esempio seguente elimina una dimensione denominata `TopicFilterForAuthMessages`.

```
aws iot delete-dimension \  
  --name TopicFilterForAuthMessages
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DeleteDimension](#) in AWS CLI Command Reference.

## **delete-domain-configuration**

Il seguente esempio di codice mostra come utilizzare `delete-domain-configuration`.

### AWS CLI

Per eliminare una configurazione di dominio

L'`delete-domain-configuration` esempio seguente elimina una configurazione di dominio denominata `additionalDataDomain` dal tuo AWS account.

```
aws iot delete-domain-configuration \  
  --domain-configuration-name "additionalDataDomain" \  
  --domain-configuration-status "OK"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configurable Endpoints](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteDomainConfiguration](#) Reference.

## **delete-dynamic-thing-group**

Il seguente esempio di codice mostra come utilizzare `delete-dynamic-thing-group`.

### AWS CLI

Per eliminare un gruppo di oggetti dinamico

L'`delete-dynamic-thing-group` esempio seguente elimina il gruppo di oggetti dinamico specificato.

```
aws iot delete-dynamic-thing-group \  
  --thing-group-name "RoomTooWarm"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Dynamic Thing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [DeleteDynamicThingGroup](#) in AWS CLI Command Reference.

## delete-job-execution

Il seguente esempio di codice mostra come utilizzare `delete-job-execution`.

### AWS CLI

Per eliminare l'esecuzione di un lavoro

L'`delete-job-execution` seguente elimina l'esecuzione del lavoro specificato su un dispositivo. Utilizzare `describe-job-execution` per ottenere il numero di esecuzione.

```
aws iot delete-job-execution
  --job-id "example-job-02"
  --thing-name "MyRaspberryPi"
  --execution-number 1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creating and Managing Jobs \(CLI\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [DeleteJobExecution AWS CLI](#) Command Reference.

## delete-job

Il seguente esempio di codice mostra come utilizzare `delete-job`.

### AWS CLI

Per eliminare un processo

L'`delete-job` seguente elimina il lavoro specificato. Specificando l'`--force` opzione, il lavoro viene eliminato anche se lo stato è `IN_PROGRESS`

```
aws iot delete-job \
  --job-id "example-job-04" \
  --force
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creating and Managing Jobs \(CLI\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [DeleteJob AWS CLI Command Reference](#).

## **delete-mitigation-action**

Il seguente esempio di codice mostra come utilizzare `delete-mitigation-action`.

AWS CLI

Per eliminare un'azione di mitigazione

L'`delete-mitigation-action` esempio seguente elimina l'azione di mitigazione specificata.

```
aws iot delete-mitigation-action \  
  --action-name AddThingsToQuarantineGroup1Action
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [DeleteMitigationAction \(Mitigation Action Commands\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DeleteMitigationAction](#) in AWS CLI Command Reference.

## **delete-ota-update**

Il seguente esempio di codice mostra come utilizzare `delete-ota-update`.

AWS CLI

Per eliminare un OTA aggiornamento

L'`delete-ota-update` esempio seguente elimina l'OTA aggiornamento specificato.

```
aws iot delete-ota-update \  
  --ota-update-id ota12345 \  
  --delete-stream \  
  --force-delete-aws-job
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [DeleteOTAUpdate](#) in AWS IoT API Reference.

- Per API i dettagli, vedere [DeleteOtaUpdate](#) in AWS CLI Command Reference.

## **delete-policy-version**

Il seguente esempio di codice mostra come utilizzare `delete-policy-version`.

### AWS CLI

Per eliminare una versione della politica

L'`delete-policy-version` esempio seguente elimina la versione 2 della politica specificata dal tuo AWS account.

```
aws iot delete-policy-version \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-version-id 2
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [AWS IoT Policies](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DeletePolicyVersion](#) in AWS CLI Command Reference.

## **delete-policy**

Il seguente esempio di codice mostra come utilizzare `delete-policy`.

### AWS CLI

Per eliminare una politica

L'`delete-policy` esempio seguente elimina la politica specificata dal tuo AWS account.

```
aws iot delete-policy --policy-name UpdateDeviceCertPolicy
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [AWS IoT Policies](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [DeletePolicy](#) in AWS CLI Command Reference.

## delete-provisioning-template-version

Il seguente esempio di codice mostra come utilizzare `delete-provisioning-template-version`.

### AWS CLI

Per eliminare una versione del modello di provisioning

L'esempio seguente elimina la versione 2 del modello di provisioning specificato.

```
aws iot delete-provisioning-template-version \  
  --version-id 2 \  
  --template-name "widget-template"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [AWS IoT Secure Tunneling](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteProvisioningTemplateVersion](#) Reference.

## delete-provisioning-template

Il seguente esempio di codice mostra come utilizzare `delete-provisioning-template`.

### AWS CLI

Per eliminare un modello di provisioning

L'esempio seguente elimina il modello di provisioning specificato.

```
aws iot delete-provisioning-template \  
  --template-name widget-template
```

Questo comando non produce alcun output.



Per ulteriori informazioni, consulta [AWS IoT Secure Tunneling](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteProvisioningTemplate](#) Reference.

## **delete-registration-code**

Il seguente esempio di codice mostra come utilizzare `delete-registration-code`.

### AWS CLI

Per eliminare il codice di registrazione

L'`delete-registration-code` esempio seguente elimina un codice di registrazione specifico AWS dell'account IoT.

```
aws iot delete-registration-code
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Use Your Own Certificate](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [DeleteRegistrationCode AWS CLI](#) Command Reference.

## **delete-role-alias**

Il seguente esempio di codice mostra come utilizzare `delete-role-alias`.

### AWS CLI

Per eliminare un alias di ruolo AWS IoT

L'`delete-role-alias` esempio seguente elimina un alias di ruolo AWS IoT denominato `LightBulbRole`

```
aws iot delete-role-alias \  
  --role-alias LightBulbRole
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Authorizing Direct Calls to AWS Services](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DeleteRoleAlias](#) in AWS CLI Command Reference.

## delete-scheduled-audit

Il seguente esempio di codice mostra come utilizzare `delete-scheduled-audit`.

### AWS CLI

Per eliminare un controllo pianificato

L'`delete-scheduled-audit` esempio seguente elimina l'audit pianificato di AWS IoT Device Defender denominato `AWSIoTDeviceDefenderDailyAudit`

```
aws iot delete-scheduled-audit \  
  --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Audit Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DeleteScheduledAudit](#) in AWS CLI Command Reference.

## delete-security-profile

Il seguente esempio di codice mostra come utilizzare `delete-security-profile`.

### AWS CLI

Per eliminare un profilo di sicurezza

L'`delete-security-profile` esempio seguente elimina un profilo di sicurezza denominato `PossibleIssue`.

```
aws iot delete-security-profile \  
  --security-profile-name PossibleIssue
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DeleteSecurityProfile](#) in AWS CLI Command Reference.

## delete-stream

Il seguente esempio di codice mostra come utilizzare `delete-stream`.

### AWS CLI

Per eliminare uno stream

L'`delete-stream`esempio seguente elimina lo stream specificato.

```
aws iot delete-stream \  
  --stream-id stream12345
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [DeleteStream](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [DeleteStream](#) in AWS CLI Command Reference.

## delete-thing-group

Il seguente esempio di codice mostra come utilizzare `delete-thing-group`.

### AWS CLI

Per eliminare un gruppo di oggetti

L'`delete-thing-group`esempio seguente elimina il gruppo di oggetti specificato. Non è possibile eliminare un gruppo di oggetti se contiene gruppi di oggetti secondari.

```
aws iot delete-thing-group \  
  --thing-group-name DefectiveBulbs
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [DeleteThingGroup](#) in AWS CLI Command Reference.

## delete-thing-type

Il seguente esempio di codice mostra come utilizzare `delete-thing-type`.

## AWS CLI

Esempio 1: Per eliminare un tipo di oggetto

L'`delete-thing-type` seguente elimina un tipo di oggetto obsoleto.

```
aws iot delete-thing-type \  
  --thing-type-name "obsoleteThingType"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Thing Types](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [DeleteThingType](#) in AWS CLI Command Reference.

## `delete-thing`

Il seguente esempio di codice mostra come utilizzare `delete-thing`.

## AWS CLI

Per visualizzare informazioni dettagliate su un oggetto

L'`delete-things` seguente elimina un elemento dal registro AWS IoT per il tuo AWS account.

```
aws iot delete-thing --thing-name» FourthBulb
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [How to Manage Things with the Registry](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [DeleteThing](#) in AWS CLI Command Reference.

## `delete-topic-rule-destination`

Il seguente esempio di codice mostra come utilizzare `delete-topic-rule-destination`.

## AWS CLI

Per eliminare la destinazione di una regola tematica

L'`delete-topic-rule-destination` seguente elimina la destinazione della regola dell'argomento specificata.

```
aws iot delete-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di una destinazione di regole tematiche](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DeleteTopicRuleDestination](#) in AWS CLI Command Reference.

## **delete-topic-rule**

Il seguente esempio di codice mostra come utilizzare `delete-topic-rule`.

AWS CLI

Per eliminare una regola

L'`delete-topic-rule` seguente elimina la regola specificata.

```
aws iot delete-topic-rule \  
  --rule-name "LowMoistureRule"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di una regola](#) nella Guida per sviluppatori AWS IoT.

- Per API i dettagli, vedere [DeleteTopicRule](#) in AWS CLI Command Reference.

## **delete-v2-logging-level**

Il seguente esempio di codice mostra come utilizzare `delete-v2-logging-level`.

AWS CLI

Per eliminare il livello di registrazione per un gruppo di oggetti

L'`delete-v2-logging-level` seguente elimina il livello di registrazione per il gruppo di oggetti specificato.

```
aws iot delete-v2-logging-level \  
  --target-type THING_GROUP \  
  --target-name LightBulbs
```

Questo comando non produce alcun output.

- Per API i dettagli, vedete [DeleteV2 in Command Reference LoggingLevel](#).AWS CLI

## deprecate-thing-type

Il seguente esempio di codice mostra come utilizzare `deprecate-thing-type`

AWS CLI

Esempio 1: Deprecare un tipo di cosa

L'`deprecate-thing-type` seguente rende obsoleto un tipo di oggetto in modo che gli utenti non possano associarvi elementi nuovi.

```
aws iot deprecate-thing-type \  
  --thing-type-name "obsoleteThingType"
```

Questo comando non produce alcun output.

Esempio 2: per invertire la deprecazione di un tipo di oggetto

L'`deprecate-thing-type` seguente inverte la deprecazione di un tipo di oggetto, in modo che gli utenti possano associarvi nuovamente elementi nuovi.

```
aws iot deprecate-thing-type \  
  --thing-type-name "obsoleteThingType" \  
  --undo-deprecate
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Thing Types](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [DeprecateThingType](#) in AWS CLI Command Reference.

## describe-account-audit-configuration

Il seguente esempio di codice mostra come utilizzare `describe-account-audit-configuration`.

### AWS CLI

Per visualizzare le impostazioni correnti della configurazione di controllo

L'esempio seguente elenca le impostazioni correnti per la configurazione di audit di AWS IoT Device Defender.

```
aws iot describe-account-audit-configuration
```

Output:

```
{
  "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
    }
  }
}

```

Per ulteriori informazioni, consulta [Audit Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DescribeAccountAuditConfiguration](#) in AWS CLI Command Reference.

## describe-audit-finding

Il seguente esempio di codice mostra come utilizzare `describe-audit-finding`.

### AWS CLI

Per elencare i dettagli relativi ai risultati di un audit

L'`describe-audit-finding` esempio seguente elenca i dettagli del risultato di audit di AWS IoT Device Defender specificato. Un audit può produrre più risultati. Usa il `list-audit-findings` comando per ottenere un elenco dei risultati di un audit per ottenere il `findingId`.

```

aws iot describe-audit-finding \
  --finding-id "ef4826b8-e55a-44b9-b460-5c485355371b"

```

Output:

```

{
  "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":
"ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS"
  }
}

```

Per ulteriori informazioni, consulta [Check Audit Results \(Audit Commands\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [DescribeAuditFinding AWS CLI Command Reference](#).

## describe-audit-mitigation-actions-task

Il seguente esempio di codice mostra come utilizzare `describe-audit-mitigation-actions-task`.

### AWS CLI

Per mostrare i dettagli di un'attività relativa alle azioni di mitigazione degli audit

L'`describe-audit-mitigation-actions-task` esempio seguente mostra i dettagli dell'attività specificata, in cui `ResetPolicyVersionAction` è stata applicata a un risultato. I risultati includono l'inizio e la fine dell'attività, il numero di risultati presi di mira (e il risultato) e la definizione dell'azione applicata come parte di questa attività.

```

aws iot describe-audit-mitigation-actions-task \
  --task-id ResetPolicyTask01

```

## Output:

```
{
  "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"
        }
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta [DescribeAuditMitigationActionsTask \(Mitigation Action Commands\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DescribeAuditMitigationActionsTask](#) in AWS CLI Command Reference.

## describe-audit-suppression

Il seguente esempio di codice mostra come utilizzare `describe-audit-suppression`.

### AWS CLI

Per ottenere dettagli sulla soppressione dei risultati di un audit

L'esempio seguente elenca i dettagli sulla soppressione di un risultato di audit.

```
aws iot describe-audit-task \  
  --task-id "787ed873b69cb4d6cdbae6ddd06996c5"
```

Output:

```
{  
  "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  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Audit finding suppressions](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeAuditSuppression](#) Reference.

## describe-audit-task

Il seguente esempio di codice mostra come utilizzare `describe-audit-task`.

### AWS CLI

Per ottenere informazioni su un'istanza di controllo

L'esempio seguente ottiene informazioni su un'istanza di un audit AWS IoT Device Defender. Se l'audit è completo, nei risultati vengono incluse le statistiche riassuntive relative all'esecuzione.

```
aws iot describe-audit-task \
  --task-id a3aea009955e501a31b764abe1bebd3d
```

Output:

```
{
  "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
    }
  }
}

```

Per ulteriori informazioni, consulta [Audit Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DescribeAuditTask](#) in AWS CLI Command Reference.

## describe-authorizer

Il seguente esempio di codice mostra come utilizzare `describe-authorizer`.

### AWS CLI

Per ottenere informazioni su un autorizzatore personalizzato

L'`describe-authorizer` esempio seguente visualizza i dettagli per l'autorizzatore personalizzato specificato.

```

aws iot describe-authorizer \
  --authorizer-name CustomAuthorizer

```

Output:

```

{
  "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-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1uJOB4lQPgG/1M6ZfIwo
\nZ+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0U8Eq9xb503kRSmyIwTzwzm/f4Gf0Y
\nZUloJ+t3PUUwHrmbYTAgrCUgRFygjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
\nzw0BkPeic0asNJpqT8PkBbRaKylJh5oo81NDHHmVtbBm5A5YiJjqYXLaVAowKzZ\n

```

```
+GqsNvAQ9Jy1wI2VrEa10fL8f1DB/BJLm7zjpfPOHDJQgID0XnZwA1NnZc0hCwIx\n50g2LW20y9R/
dmqtDmJiVP97Z4GykxPvwLYHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1\nlQIDAQAB\n-----END PUBLIC
KEY-----"
    },
    "status": "ACTIVE",
    "creationDate": 1571245658.069,
    "lastModifiedDate": 1571245658.069
  }
}
```

Per ulteriori informazioni, [DescribeAuthorizer](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [DescribeAuthorizer](#) in AWS CLI Command Reference.

## describe-billing-group

Il seguente esempio di codice mostra come utilizzare `describe-billing-group`.

### AWS CLI

Per ottenere informazioni su un gruppo di fatturazione

L'`describe-billing-group` esempio seguente ottiene informazioni per il gruppo di fatturazione specificato.

```
aws iot describe-billing-group --billing-group-name GroupOne
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Billing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [DescribeBillingGroup AWS CLI Command Reference](#).

## describe-ca-certificate

Il seguente esempio di codice mostra come utilizzare `describe-ca-certificate`.

### AWS CLI

Per ottenere dettagli su un certificato CA

L'esempio seguente visualizza i dettagli del certificato CA specificato.

```
aws iot describe-ca-certificate \
  --certificate-
  id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467
```

Output:

```
{
  "certificateDescription": {
    "certificateArn": "arn:aws:iot:us-west-2:123456789012:ca-cert/
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
    "certificateId":
"f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
    "status": "INACTIVE",
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIICzzCCAbegEXAMPLEJANVEPWX18taPMA0GCSqGSIb3DQEBBQUAMB4xCzAJBgNV
\nBAYTA1VMTMQ8wDQYDVQQKDAZBbWF6b24wHhcNMTEwMjEzMTU1WhcNMjEzMTU1
\nMjEzMTU1WjAeMQswCQYDVQQGEwJVUzEPMA0GA1UECgwW1hem9uMIIBIjANBgkq
\nhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAZd3R3ioa1CS0MhFWfBrVGR036EK07UAF
\nVdz9EXAMPLE1VczICbADnATK522kEIB51/18Vz1FtAhQL5V5eybXKnB7QebNer5m
\n4Yibx7shR5oqNzFsrXWxuugN5+w5gEfqNMaw0jhF4Lscu1KG49yuqjcDU19/13ua
\n3B2gxs1Pe7TiWwvUskzxbn01F2WCshbEJvqY8fIwtGYCjTeJAgQ9hvZx/69XhKen
\nwV9LJw0QxrsUS0Ty8IHwbB8fRy72VM3u7fJoaU+n04jd5cqaoEPtzoEEXAMPLE
\nyVAJpqHwgbYbcUfn7V+AB6yh1+0Fa1rEQGuZDPGyJslxwr5vh8nRewIDAQABoxAw
\nDjAMBgNVHRMERTADAQH/MA0GCSqGSIb3DQEBBQUAA4IBAQA+3a5CV3IJg0nd0AgI
\nBgVMtmYzTvqAngx26aG9/spvCjXckh2SBF+EcB1CFwH1yakwjJL1dR4yarnrfxgI
\nEqP4A0YVimAVoQ5FBwnloHe16+3qtDib1U9DeXBUctS55EcfEXAMPLEYtXdqU5C
\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
    }
  }
}

```

Per ulteriori informazioni, vedere [D escriveCACertificate](#) in AWS IoT API Reference.

- Per API i dettagli, vedere [DescribeCaCertificate](#) in AWS CLI Command Reference.

## describe-certificate

Il seguente esempio di codice mostra come utilizzare `describe-certificate`.

### AWS CLI

Per ottenere informazioni su un certificato

L'`describe-certificate` esempio seguente visualizza i dettagli del certificato specificato.

```

aws iot describe-certificate \
  --certificate-
  id "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"

```

Output:

```

{
  "certificateDescription": {
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
    "certificateId":
    "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
    "status": "ACTIVE",
    "certificatePem": "-----BEGIN CERTIFICATE-----
MIICiTEXAMPLEQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBEXAMPLEMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDEXAMPLElMRIwEAYDVQQDEw1UZXR0Q21sYWxhZAd
BgkqhkiG9w0BCQEWEG5vb251QGFTYXpvbi5EXAMPLEcNMTEwNDI1MjA0NTIxWhcN

```



```

MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCMVVMxCzAJBgNEXAMPLEdBMRAdGgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC01BEXAMPLEz
b2xEXAMPLEYDVQDEw1UZXR0Q21sYWxhZAdBgkqhkiG9w0BCQEWEG5vb251QGft
YXpvbi5jb20wgZ8EXAMPLEZIHvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTc2XADZ4nB+BLYEXAMPLEpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7EXAMPLEGBzZswY6786m86gpE
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFEXAMPLEAtCu4
nUhVVxYUnEXAMPLE8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GEXAMPLEl0ZxBHjJnyp3780D8uTs7fLvJx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
    "ownedBy": "123456789012",
    "creationDate": 1541022751.983,
    "lastModifiedDate": 1541022751.983,
    "customerVersion": 1,
    "transferData": {},
    "generationId": "6974fbed-2e61-4114-bc5e-4204cc79b045",
    "validity": {
      "notBefore": 1541022631.0,
      "notAfter": 2524607999.0
    }
  }
}
}

```

Per ulteriori informazioni, [DescribeCertificate](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [DescribeCertificate](#) in AWS CLI Command Reference.

## describe-custom-metric

Il seguente esempio di codice mostra come utilizzare `describe-custom-metric`.

### AWS CLI

Per ottenere informazioni su una metrica personalizzata di Device Defender

L'`describe-custom-metric` esempio seguente ottiene informazioni su una metrica personalizzata denominata `myCustomMetric`

```

aws iot describe-custom-metric \
  --metric-name myCustomMetric

```

Output:

```
{
  "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
}
```

Per ulteriori informazioni, consulta [Metriche personalizzate](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [DescribeCustomMetric AWS CLI Command Reference](#).

## **describe-default-authorizer**

Il seguente esempio di codice mostra come utilizzare `describe-default-authorizer`.

### AWS CLI

Per ottenere informazioni sull'autorizzatore personalizzato predefinito

L'`describe-default-authorizer` seguente visualizza i dettagli per l'autorizzatore personalizzato predefinito.

```
aws iot describe-default-authorizer
```

Output:

```
{
  "authorizerName": "CustomAuthorizer",
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/CustomAuthorizer"
}
```

Per ulteriori informazioni, [DescribeDefaultAuthorizer](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [DescribeDefaultAuthorizer](#) in AWS CLI Command Reference.

## **describe-dimension**

Il seguente esempio di codice mostra come utilizzare `describe-dimension`.

## AWS CLI

Per ottenere informazioni su una dimensione

L'`describe-dimension`esempio seguente ottiene informazioni su una dimensione denominata `TopicFilterForAuthMessages`.

```
aws iot describe-dimension \  
  --name TopicFilterForAuthMessages
```

Output:

```
{  
  "name": "TopicFilterForAuthMessages",  
  "arn": "arn:aws:iot:eu-west-2:123456789012:dimension/  
TopicFilterForAuthMessages",  
  "type": "TOPIC_FILTER",  
  "stringValues": [  
    "device/+/auth"  
  ],  
  "creationDate": 1578620223.255,  
  "lastModifiedDate": 1578620223.255  
}
```

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DescribeDimension](#) in AWS CLI Command Reference.

## `describe-domain-configuration`

Il seguente esempio di codice mostra come utilizzare `describe-domain-configuration`.

## AWS CLI

Per descrivere una configurazione di dominio

L'`describe-domain-configuration`esempio seguente visualizza i dettagli sulla configurazione del dominio specificata.

```
aws iot describe-domain-configuration \  
  --domain-configuration-name "additionalDataDomain"
```

**Output:**

```
{
  "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
}
```

Per ulteriori informazioni, consulta [Configurable Endpoints](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeDomainConfiguration](#) Reference.

**describe-endpoint**

Il seguente esempio di codice mostra come utilizzare `describe-endpoint`.

**AWS CLI**

Esempio 1: Per ottenere l' AWS endpoint corrente

L'`describe-endpoint` esempio seguente recupera l' AWS endpoint predefinito a cui vengono applicati tutti i comandi.

```
aws iot describe-endpoint
```

**Output:**

```
{
  "endpointAddress": "abc123defghijk.iot.us-west-2.amazonaws.com"
}
```

Per ulteriori informazioni, consulta [DescribeEndpoint](#) nella AWS IoT Developer Guide.

Esempio 2: Per ottenere il tuo ATS endpoint

L'`describe-endpoint` esempio seguente recupera l'endpoint Amazon Trust Services (ATS).

```
aws iot describe-endpoint \  
  --endpoint-type iot:Data-ATS
```

Output:

```
{  
  "endpointAddress": "abc123defghijk-ats.iot.us-west-2.amazonaws.com"  
}
```

Per ulteriori informazioni, consulta [Certificati X.509 e IoT nella AWS IoT Developer Guide](#).

- Per API i dettagli, consulta AWS CLI Command [DescribeEndpoint](#) Reference.

## describe-event-configurations

Il seguente esempio di codice mostra come utilizzare `describe-event-configurations`.

AWS CLI

Per mostrare quali tipi di eventi sono pubblicati

L'`describe-event-configurations` seguente elenca la configurazione che controlla quali eventi vengono generati quando qualcosa viene aggiunto, aggiornato o eliminato.

```
aws iot describe-event-configurations
```

Output:

```
{  
  "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
    }
  }
}

```

Per ulteriori informazioni, consulta [Event Messages](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DescribeEventConfigurations](#) in AWS CLI Command Reference.

## describe-index

Il seguente esempio di codice mostra come utilizzare `describe-index`.

### AWS CLI

Per recuperare lo stato corrente dell'indice della cosa

L'`describe-index` esempio seguente recupera lo stato corrente dell'indice di cosa.

```

aws iot describe-index \
  --index-name "AWS_Things"

```

Output:

```
{
  "indexName": "AWS_Things",
  "indexStatus": "ACTIVE",
  "schema": "REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS"
}
```

Per ulteriori informazioni, consulta [Managing Thing Indexing](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DescribeIndex](#) in AWS CLI Command Reference.

## describe-job-execution

Il seguente esempio di codice mostra come utilizzare `describe-job-execution`.

### AWS CLI

Per ottenere i dettagli di esecuzione di un processo su un dispositivo

L'`describe-job-execution` seguente ottiene i dettagli di esecuzione per il processo specificato.

```
aws iot describe-job-execution \
  --job-id "example-job-01" \
  --thing-name "MyRaspberryPi"
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Creating and Managing Jobs \(CLI\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [DescribeJobExecution AWS CLI Command Reference](#).

## describe-job

Il seguente esempio di codice mostra come utilizzare `describe-job`.

### AWS CLI

Per ottenere lo stato dettagliato di un lavoro

L'`describe-job`esempio seguente ottiene lo stato dettagliato del lavoro il cui ID è `example-job-01`.

```
aws iot describe-job \  
  --job-id "example-job-01"
```

Output:

```
{  
  "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": {}
  }
}

```

Per ulteriori informazioni, consulta [Creating and Managing Jobs \(CLI\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [DescribeJob AWS CLI Command Reference](#).

## describe-mitigation-action

Il seguente esempio di codice mostra come utilizzare `describe-mitigation-action`.

### AWS CLI

Per visualizzare i dettagli di un'azione di mitigazione definita

L'`describe-mitigation-action` esempio seguente visualizza i dettagli dell'azione di mitigazione specificata.

```

aws iot describe-mitigation-action \
  --action-name AddThingsToQuarantineGroupAction

```

Output:

```

{
  "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"
}
```

Per ulteriori informazioni, consulta [DescribeMitigationAction \(Mitigation Action Commands\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DescribeMitigationAction](#) in AWS CLI Command Reference.

## describe-provisioning-template-version

Il seguente esempio di codice mostra come utilizzare `describe-provisioning-template-version`.

### AWS CLI

Per descrivere una versione del modello di provisioning

L'`describe-provisioning-template-version`esempio seguente descrive una versione del modello di provisioning.

```
aws iot describe-provisioning-template-version \
  --template-name MyTestProvisioningTemplate \
  --version-id 1
```

Output:

```
{
  "versionId": 1,
  "creationDate": 1589308310.574,
  "templateBody": "{
    \"Parameters\":{
      \"SerialNumber\":{
        \"Type\":\"String\"
      },
      \"AWS::IoT::Certificate::Id\":{
        \"Type\":\"String\"
      }
    },
    \"Resources\":{
      \"certificate\":{
        \"Properties\":{
          \"CertificateId\":{
```



## describe-provisioning-template

Il seguente esempio di codice mostra come utilizzare `describe-provisioning-template`.

### AWS CLI

Per descrivere un modello di provisioning

L'esempio seguente descrive un modello di provisioning.

```
aws iot describe-provisioning-template \  
  --template-name MyTestProvisioningTemplate
```

Output:

```
{  
  "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\"
}

```

Per ulteriori informazioni, consulta [Provisioning di dispositivi che non dispongono di certificati di dispositivo utilizzando il fleet provisioning](#) nella AWS IoT Core Developers Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeProvisioningTemplate](#) Reference.

## describe-role-alias

Il seguente esempio di codice mostra come utilizzare `describe-role-alias`.

### AWS CLI

Per ottenere informazioni su un alias di ruolo AWS IoT

L'`describe-role-alias` esempio seguente visualizza i dettagli per l'alias del ruolo specificato.

```
aws iot describe-role-alias \  
  --role-alias LightBulbRole
```

Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, [DescribeRoleAlias](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [DescribeRoleAlias](#) in AWS CLI Command Reference.

## `describe-scheduled-audit`

Il seguente esempio di codice mostra come utilizzare `describe-scheduled-audit`.

AWS CLI

Per ottenere informazioni su un audit pianificato

L'`describe-scheduled-audit` esempio seguente ottiene informazioni dettagliate su un audit pianificato di AWS IoT Device Defender denominato `AWSIoTDeviceDefenderDailyAudit`.

```
aws iot describe-scheduled-audit \  
  --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```

Output:

```
{
```

```

    "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"
  }

```

Per ulteriori informazioni, consulta [Audit Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DescribeScheduledAudit](#) in AWS CLI Command Reference.

## describe-security-profile

Il seguente esempio di codice mostra come utilizzare `describe-security-profile`.

### AWS CLI

Per ottenere informazioni su un profilo di sicurezza

L'esempio seguente ottiene informazioni sul profilo di sicurezza AWS IoT Device Defender denominato `PossibleIssue`.

```

aws iot describe-security-profile \
  --security-profile-name PossibleIssue

```

Output:

```

{
  "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
}
```

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DescribeSecurityProfile](#) in AWS CLI Command Reference.

## describe-stream

Il seguente esempio di codice mostra come utilizzare `describe-stream`.

### AWS CLI

Per ottenere informazioni su uno stream

L'esempio seguente mostra i dettagli sullo stream specificato.



```
aws iot describe-stream \  
  --stream-id stream12345
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, [DescribeStream](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [DescribeStream](#) in AWS CLI Command Reference.

## describe-thing-group

Il seguente esempio di codice mostra come utilizzare `describe-thing-group`.

AWS CLI

Per ottenere informazioni su un gruppo di oggetti

L'esempio seguente ottiene informazioni sul gruppo di cose denominato `HalogenBulbs`.

```
aws iot describe-thing-group \  
  --thing-group-name HalogenBulbs
```

```
--thing-group-name HalogenBulbs
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [DescribeThingGroup](#) in AWS CLI Command Reference.

## describe-thing-type

Il seguente esempio di codice mostra come utilizzare `describe-thing-type`.

AWS CLI

Per ottenere informazioni su un oggetto, digita

L'esempio seguente mostra informazioni sul tipo di oggetto specificato definito nell'AWS account.

```
aws iot describe-thing-type \  
  --thing-type-name "LightBulb"
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Thing Types](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [DescribeThingType](#) in AWS CLI Command Reference.

## describe-thing

Il seguente esempio di codice mostra come utilizzare `describe-thing`.

### AWS CLI

Per visualizzare informazioni dettagliate su un oggetto

L'esempio seguente mostra informazioni su un oggetto (dispositivo) definito nel registro AWS IoT per l'AWS account.

```
aws iot describe-thing --thing-name "» MyLightBulb
```

Output:

```
{
  "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
}
```

Per ulteriori informazioni, consulta [How to Manage Things with the Registry](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [DescribeThing](#) in AWS CLI Command Reference.

## detach-policy

Il seguente esempio di codice mostra come utilizzare `detach-policy`.

### AWS CLI

Esempio 1: scollegare una policy AWS IoT da un gruppo di oggetti

L'`detach-policy` esempio seguente separa la policy specificata da un gruppo di cose e, per estensione, da tutti gli elementi di quel gruppo e da qualsiasi gruppo figlio del gruppo.

```
aws iot detach-policy \  
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \  
  --policy-name "MyFirstGroup_Core-policy"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developers Guide.

Esempio 2: scollegare una policy AWS IoT da un certificato del dispositivo

L'`detach-policy` esempio seguente separa la `TemperatureSensorPolicy` policy da un certificato di dispositivo identificato da ARN

```
aws iot detach-policy \  
  --policy-name TemperatureSensorPolicy \  
  --target arn:aws:iot:us-  
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DetachPolicy](#) in AWS CLI Command Reference.

## detach-security-profile

Il seguente esempio di codice mostra come utilizzare `detach-security-profile`.

### AWS CLI

Per dissociare un profilo di sicurezza da una destinazione

L'esempio seguente rimuove l'associazione tra il profilo di sicurezza AWS IoT Device Defender denominato `Testprofile` e l'oggetto `All Registered Things`.

```
aws iot detach-security-profile \  
  --security-profile-name Testprofile \  
  --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/  
registered-things"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DetachSecurityProfile](#) in AWS CLI Command Reference.

## detach-thing-principal

Il seguente esempio di codice mostra come utilizzare `detach-thing-principal`.

### AWS CLI

Scollegare un certificato/principale da un oggetto

L'esempio seguente rimuove un certificato che rappresenta un principale dall'oggetto specificato.

```
aws iot detach-thing-principal \  
  --thing-name "MyLightBulb" \  
  --principal "arn:aws:iot:us-  
west-2:123456789012:cert/604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [How to Manage Things with the Registry](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [DetachThingPrincipal](#) in AWS CLI Command Reference.

## **disable-topic-rule**

Il seguente esempio di codice mostra come utilizzare `disable-topic-rule`.

### AWS CLI

Per disabilitare una regola tematica

L'`disable-topic-rule` esempio seguente disattiva la regola dell'argomento specificata.

```
aws iot disable-topic-rule \  
  --rule-name "MyPlantPiMoistureAlertRule"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Viewing Your Rules](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [DisableTopicRule AWS CLI](#) Command Reference.

## **enable-topic-rule**

Il seguente esempio di codice mostra come utilizzare `enable-topic-rule`.

### AWS CLI

Per abilitare una regola tematica

L'`enable-topic-rule` esempio seguente abilita (o riattiva) la regola dell'argomento specificata.

```
aws iot enable-topic-rule \  
  --rule-name "MyPlantPiMoistureAlertRule"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Viewing Your Rules](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [EnableTopicRule AWS CLI Command Reference](#).

## get-behavior-model-training-summaries

Il seguente esempio di codice mostra come utilizzare `get-behavior-model-training-summaries`.

### AWS CLI

Per elencare lo stato del modello di formazione ML Detect Security Profile di Device Defender

L'`get-behavior-model-training-summaries`esempio seguente elenca lo stato di addestramento del modello per i comportamenti configurati nel profilo di sicurezza scelto. Per ogni comportamento, vengono elencati il nome, lo stato del modello e la percentuale di datapoint raccolti.

```
aws iot get-behavior-model-training-summaries \  
  --security-profile-name MySecuirtyProfileName
```

Output:

```
{  
  "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
    }
  ]
}

```

Per ulteriori informazioni, consulta [GetBehaviorModelTrainingSummaries \(Detect Commands\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [GetBehaviorModelTrainingSummaries](#) in AWS CLI Command Reference.

## get-cardinality

Il seguente esempio di codice mostra come utilizzare `get-cardinality`.

### AWS CLI

Per restituire il conteggio approssimativo dei valori univoci che corrispondono alla query

È possibile utilizzare il seguente script di configurazione per creare 10 elementi che rappresentano 10 sensori di temperatura. Ogni cosa nuova ha 3 attributi.

```

# 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
```

Esempio di output dello script di configurazione:

```
{
  "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"
}
```

L'get-cardinality esempio seguente interroga i 10 sensori creati dallo script di configurazione e restituisce il numero di rack in cui i sensori di temperatura segnalano valori di temperatura anomali. Se il valore della temperatura è inferiore a 60 o superiore a 80, il sensore di temperatura si trova in uno stato anomalo.

```
aws iot get-cardinality \
  --aggregation-field "attributes.rackId" \
  --query-string "thingName:TempSensor* AND attributes.stateNormal:false"
```

Output:

```
{
  "cardinality": 2
}
```

Per ulteriori informazioni, consulta Querying for Aggregate Data < <https://docs.aws.amazon.com/iot/latest/developerguide/index-aggregate.html> > nella IoT Developer Guide.AWS

- Per i API AWS CLI dettagli, [GetCardinality](#) vedere in Command Reference.

## get-effective-policies

Il seguente esempio di codice mostra come utilizzare `get-effective-policies`.

### AWS CLI

Elencare le politiche che influiscono su qualcosa

L'`get-effective-policies` esempio seguente elenca le politiche che influiscono sull'elemento specificato, incluse le politiche associate a tutti i gruppi a cui appartiene.

```
aws iot get-effective-policies \  
  --thing-name TemperatureSensor-001 \  
  --principal arn:aws:iot:us-west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

Output:

```
{  
  "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
\ "
        ]
    }
]
}"
}
]
}

```

Per ulteriori informazioni, consulta [Get Effective Policies for a Thing](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [GetEffectivePolicies AWS CLI Command Reference](#).

## get-indexing-configuration

Il seguente esempio di codice mostra come utilizzare `get-indexing-configuration`.

### AWS CLI

Per ottenere la configurazione di indicizzazione degli oggetti

L'esempio seguente ottiene i dati di configurazione correnti per l'indicizzazione della flotta AWS IoT.

```
aws iot get-indexing-configuration
```

**Output:**

```
{
  "thingIndexingConfiguration": {
    "thingIndexingMode": "OFF",
    "thingConnectivityIndexingMode": "OFF"
  },
  "thingGroupIndexingConfiguration": {
    "thingGroupIndexingMode": "OFF"
  }
}
```

Per ulteriori informazioni, consulta [Managing Thing Indexing](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [GetIndexingConfiguration](#) in AWS CLI Command Reference.

**get-job-document**

Il seguente esempio di codice mostra come utilizzare `get-job-document`.

**AWS CLI**

Per recuperare il documento per un lavoro

L'esempio seguente visualizza i dettagli sul documento relativo al lavoro il cui ID è `example-job-01`.

```
aws iot get-job-document \
  --job-id "example-job-01"
```

**Output:**

```
{
  "document": "\n{\n  \"operation\": \"customJob\", \n  \"otherInfo\":\n  \"someValue\"\n}\n"
```

Per ulteriori informazioni, consulta [Creating and Managing Jobs \(CLI\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [GetJobDocument AWS CLI Command Reference](#).

## get-logging-options

Il seguente esempio di codice mostra come utilizzare `get-logging-options`.

### AWS CLI

Per ottenere le opzioni di registrazione

L'`get-logging-options`esempio seguente ottiene le opzioni di registrazione correnti per il tuo AWS account.

```
aws iot get-logging-options
```

Output:

```
{
  "roleArn": "arn:aws:iam::123456789012:role/service-role/iotLoggingRole",
  "logLevel": "ERROR"
}
```

Per ulteriori informazioni, consulta il titolo della AWS IoT Developer Guide.

- Per API i dettagli, vedere [GetLoggingOptions](#) in AWS CLI Command Reference.

## get-ota-update

Il seguente esempio di codice mostra come utilizzare `get-ota-update`.

### AWS CLI

Per recuperare informazioni su un aggiornamento OTA

L'`get-ota-update`esempio seguente visualizza i dettagli sull'OTAaggiornamento specificato.

```
aws iot get-ota-update \
  --ota-update-id ota12345
```

Output:

```
{
```

```
"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": {
  }
}
}
```

Per ulteriori informazioni, consulta [GetOTAUpdate](#) in AWS IoT API Reference.

- Per API i dettagli, vedere [GetOtaUpdate](#) in AWS CLI Command Reference.

## get-percentiles

Il seguente esempio di codice mostra come utilizzare `get-percentiles`.

### AWS CLI

Per raggruppare i valori aggregati che corrispondono alla query in raggruppamenti percentili

È possibile utilizzare il seguente script di configurazione per creare 10 elementi che rappresentano 10 sensori di temperatura. Ogni cosa nuova ha 1 attributo.

```
# 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
```

Esempio di output dello script di configurazione:

```
{
  "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"
}
```

L'`get-percentiles` seguente interroga i 10 sensori creati dallo script di configurazione e restituisce un valore per ogni gruppo di percentili specificato. Il gruppo percentile «10» contiene il valore del campo aggregato che si trova in circa il 10 per cento dei valori che corrispondono alla query. Nell'output seguente, `{"percent": 10.0, «value»: 67,7}` significa che circa il 10,0% dei valori di temperatura è inferiore a 67,7.

```
aws iot get-percentiles \
  --aggregation-field "attributes.temperature" \
  --query-string "thingName:TempSensor*" \
```

```
--percentils 10 25 50 75 90
```

Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta [Querying for Aggregate Data](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetPercentiles](#) Reference.

## get-policy-version

Il seguente esempio di codice mostra come utilizzare `get-policy-version`.

AWS CLI

Per ottenere informazioni su una versione specifica di una politica

L'esempio seguente ottiene informazioni sulla prima versione della politica specificata.



```
aws iot get-policy \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-version-id "1"
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [AWS IoT Policies](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [GetPolicyVersion](#) in AWS CLI Command Reference.

## get-policy

Il seguente esempio di codice mostra come utilizzare `get-policy`.

### AWS CLI

Per ottenere informazioni sulla versione predefinita di una politica

L'esempio seguente recupera informazioni sulla versione predefinita della politica specificata.

```
aws iot get-policy \  
  --policy-name UpdateDeviceCertPolicy
```

Output:

```
{  
  "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"
}
```

Per ulteriori informazioni, consulta [AWS IoT Policies](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [GetPolicy](#) in AWS CLI Command Reference.

## get-registration-code

Il seguente esempio di codice mostra come utilizzare `get-registration-code`.

### AWS CLI

Per ottenere il codice di registrazione AWS specifico per l'account

L'`get-registration-code` esempio seguente recupera il codice di registrazione specifico dell'AWS account.

```
aws iot get-registration-code
```

Output:

```
{
  "registrationCode":
  "15c51ae5e36ba59ba77042df1115862076bea4bd15841c838fcb68d5010a614c"
}
```

Per ulteriori informazioni, consulta [Use Your Own Certificate](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [GetRegistrationCode AWS CLI](#) Command Reference.

## get-statistics

Il seguente esempio di codice mostra come utilizzare `get-statistics`.

## AWS CLI

Per cercare dati aggregati nell'indice del dispositivo

L'`get-statistics` esempio seguente restituisce il numero di elementi che hanno una proprietà chiamata `connectivity.connected` set to `false` (ovvero il numero di dispositivi che non sono connessi) nell'ombra del dispositivo.

```
aws iot get-statistics \  
  --index-name AWS_Things \  
  --query-string "connectivity.connected:false"
```

Output:

```
{  
  "statistics": {  
    "count": 6  
  }  
}
```

Per ulteriori informazioni, consulta [Ottenerne statistiche sulla flotta di dispositivi](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [GetStatistics AWS CLI](#) Command Reference.

## `get-topic-rule-destination`

Il seguente esempio di codice mostra come utilizzare `get-topic-rule-destination`.

## AWS CLI

Per ottenere una destinazione per le regole tematiche

L'`get-topic-rule-destination` esempio seguente ottiene informazioni sulla destinazione di una regola tematica.

```
aws iot get-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

Output:

```
{
  "topicRuleDestination": {
    "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "status": "DISABLED",
    "httpUrlProperties": {
      "confirmationUrl": "https://example.com"
    }
  }
}
```

Per ulteriori informazioni, consulta [Lavorare con le destinazioni delle regole tematiche](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [GetTopicRuleDestination AWS CLI Command Reference](#).

## get-topic-rule

Il seguente esempio di codice mostra come utilizzare `get-topic-rule`.

### AWS CLI

Per ottenere informazioni su una regola

L'`get-topic-rule` esempio seguente ottiene informazioni sulla regola specificata.

```
aws iot get-topic-rule \
  --rule-name MyRPiLowMoistureAlertRule
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Viewing Your Rules](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [GetTopicRule](#) in AWS CLI Command Reference.

## get-v2-logging-options

Il seguente esempio di codice mostra come utilizzare `get-v2-logging-options`.

### AWS CLI

Per elencare le opzioni di registrazione correnti

L'`get-v2-logging-options` seguente elenca le opzioni di registrazione correnti per AWS IoT.

```
aws iot get-v2-logging-options
```

Output:

```
{
  "roleArn": "arn:aws:iam::094249569039:role/service-role/iotLoggingRole",
  "defaultLogLevel": "WARN",
  "disableAllLogs": false
}
```

Per ulteriori informazioni, consulta il titolo della AWS IoT Developer Guide.

- Per API i dettagli, consulta [GetV2 LoggingOptions](#) in AWS CLI Command Reference.

## list-active-violations

Il seguente esempio di codice mostra come utilizzare `list-active-violations`

### AWS CLI

Per elencare le violazioni attive

L'`list-active-violations` seguente elenca tutte le violazioni per il profilo di sicurezza specificato.

```
aws iot list-active-violations \  
  --security-profile-name Testprofile
```

Output:

```
{  
  "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": "c8a9466a093d3b7b35cd44ca58bdbbeab",
```

```
    "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
  }
]
}

```

- Per API i dettagli, vedere [ListActiveViolations](#) in AWS CLI Command Reference.

## list-attached-policies

Il seguente esempio di codice mostra come utilizzare `list-attached-policies`.

### AWS CLI

Esempio 1: Per elencare le politiche allegate a un gruppo

L'`list-attached-policies` esempio seguente elenca le politiche allegate al gruppo specificato.

```

aws iot list-attached-policies \
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"

```

Output:



```
{
  "policies": [
    {
      "policyName": "UpdateDeviceCertPolicy",
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
UpdateDeviceCertPolicy"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developers Guide.

Esempio 2: per elencare le politiche allegate al certificato di un dispositivo

L'`list-attached-policies` seguente elenca le politiche AWS IoT allegate al certificato del dispositivo. Il certificato è identificato dal suo ARN.

```
aws iot list-attached-policies \
  --target arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

Output:

```
{
  "policies": [
    {
      "policyName": "TemperatureSensorPolicy",
      "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [ListAttachedPolicies](#) in AWS CLI Command Reference.

## list-audit-findings

Il seguente esempio di codice mostra come utilizzare `list-audit-findings`.

## AWS CLI

Esempio 1: elencare tutti i risultati di un audit

L'`list-audit-findings` seguente elenca tutti i risultati di un audit AWS IoT Device Defender con un ID attività specificato.

```
aws iot list-audit-findings \  
  --task-id a3aea009955e501a31b764abe1bebd3d
```

Output:

```
{  
  "findings": []  
}
```

Esempio 2: Per elencare i risultati di un controllo, digita

L'`list-audit-findings` seguente mostra i risultati degli audit di AWS IoT Device Defender eseguiti tra il 5 giugno 2019 e il 19 giugno 2019 in cui i dispositivi condividono un certificato del dispositivo. Quando si specifica un nome per l'assegno, è necessario fornire un'ora di inizio e di fine.

```
aws iot list-audit-findings \  
  --check-name DEVICE_CERTIFICATE_SHARED_CHECK \  
  --start-time 1559747125 \  
  --end-time 1560962028
```

Output:

```
{  
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Audit Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ListAuditFindings](#) in AWS CLI Command Reference.

## **list-audit-mitigation-actions-executions**

Il seguente esempio di codice mostra come utilizzare `list-audit-mitigation-actions-executions`.

### AWS CLI

Per elencare i dettagli dell'esecuzione di un'azione di mitigazione degli audit

Un'attività di mitigazione degli audit applica un'azione di mitigazione a uno o più risultati di un audit di AWS IoT Device Defender. L'`list-audit-mitigation-actions-executions` esempio

seguente elenca i dettagli dell'attività di mitigazione con il risultato specificato `taskId` e per il risultato specificato.

```
aws iot list-audit-mitigation-actions-executions \  
  --task-id myActionsTaskId \  
  --finding-id 0edbaaec-2fe1-4cf5-abc9-d4c3e51f7464
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [ListAuditMitigationActionsExecutions \(Mitigation Action Commands\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ListAuditMitigationActionsExecutions](#) in AWS CLI Command Reference.

## **list-audit-mitigation-actions-tasks**

Il seguente esempio di codice mostra come utilizzare `list-audit-mitigation-actions-tasks`.

AWS CLI

Per elencare le attività relative alle azioni di mitigazione degli audit

L'`list-audit-mitigation-actions-tasks`esempio seguente elenca le azioni di mitigazione che sono state applicate ai risultati entro il periodo di tempo specificato.

```
aws iot list-audit-mitigation-actions-tasks \  
  --start-time 1594157400 \  
  --end-time 1594157400
```

```
--end-time 1594157430
```

Output:

```
{
  "tasks": [
    {
      "taskId": "0062f2d6-3999-488f-88c7-bef005414103",
      "startTime": "2020-07-07T14:30:15.172000-07:00",
      "taskStatus": "COMPLETED"
    }
  ]
}
```

Per ulteriori informazioni, consulta [ListAuditMitigationActionsTasks \(Mitigation Action Commands\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ListAuditMitigationActionsTasks](#) in AWS CLI Command Reference.

## list-audit-suppressions

Il seguente esempio di codice mostra come utilizzare `list-audit-suppressions`.

AWS CLI

Per elencare tutte le soppressioni dei risultati degli audit

L'`list-audit-suppressions` esempio seguente elenca tutte le soppressioni attive dei risultati di audit.

```
aws iot list-audit-suppressions
```

Output:

```
{
  "suppressions": [
    {
      "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
      "resourceIdentifier": {
        "deviceCertificateId": "c7691e<shortened>"
      },
      "expirationDate": 1597881600.0,
    }
  ]
}
```

```
    "suppressIndefinitely": false
  }
]
}
```

Per ulteriori informazioni, consulta [Audit finding suppressions](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta AWS CLI Command [ListAuditSuppressionsReference](#).

## list-audit-tasks

Il seguente esempio di codice mostra come utilizzare `list-audit-tasks`.

### AWS CLI

Per elencare tutti i risultati di un audit

L'`list-audit-tasks` esempio seguente elenca le attività di audit eseguite tra il 5 giugno 2019 e il 12 giugno 2019.

```
aws iot list-audit-tasks \
  --start-time 1559747125 \
  --end-time 1560357228
```

Output:

```
{
  "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": "eeef61068b0eb03c456d746c5a26ee04",
  "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"
}
]
```

Per ulteriori informazioni, consulta [Audit Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ListAuditTasks](#) in AWS CLI Command Reference.

## list-authorizers

Il seguente esempio di codice mostra come utilizzare `list-authorizers`.

### AWS CLI

Per elencare il tuo autorizzatore personalizzato

L'`list-authorizers` esempio seguente elenca gli autorizzatori personalizzati presenti nel tuo AWS account.

```
aws iot list-authorizers
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, [ListAuthorizers](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [ListAuthorizers](#) in AWS CLI Command Reference.

## **list-billing-groups**

Il seguente esempio di codice mostra come utilizzare `list-billing-groups`.

AWS CLI

Per elencare i gruppi di fatturazione per il tuo AWS account e la tua regione

L'`list-billing-groups` esempio seguente elenca tutti i gruppi di fatturazione definiti per l'AWS account e AWS la regione.

```
aws iot list-billing-groups
```

Output:

```
{
  "billingGroups": [
    {
      "groupName": "GroupOne",
      "groupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Billing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [ListBillingGroups AWS CLI Command Reference](#).

## list-ca-certificates

Il seguente esempio di codice mostra come utilizzare `list-ca-certificates`.

AWS CLI

Per elencare i certificati CA registrati nel tuo AWS account

L'`list-ca-certificates` esempio seguente elenca i certificati CA registrati nel tuo AWS account.

```
aws iot list-ca-certificates
```

Output:

```
{
  "certificates": [
    {
      "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
      "certificateId":
"f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
      "status": "INACTIVE",
    }
  ]
}
```

```

        "creationDate": 1569365372.053
      }
    ]
  }

```

Per ulteriori informazioni, consulta [Use Your Own Certificate](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [ListCaCertificates AWS CLI Command Reference](#).

## list-certificates-by-ca

Il seguente esempio di codice mostra come utilizzare `list-certificates-by-ca`.

### AWS CLI

Per elencare tutti i certificati dei dispositivi firmati con un certificato CA

L'`list-certificates-by-ca` esempio seguente elenca tutti i certificati dei dispositivi presenti nell' AWS account che sono firmati con il certificato CA specificato.

```

aws iot list-certificates-by-ca \
  --ca-certificate-
  id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467

```

Output:

```

{
  "certificates": [
    {
      "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "certificateId":
"488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "status": "ACTIVE",
      "creationDate": 1569363250.557
    }
  ]
}

```

Per ulteriori informazioni, consulta [ListCertificatesByCA](#) nell'AWS IoT API Reference.

- Per API i dettagli, consulta [ListCertificatesByCa AWS CLI Command Reference](#).

## list-certificates

Il seguente esempio di codice mostra come utilizzare `list-certificates`.

### AWS CLI

Esempio 1: Per elencare i certificati registrati nel tuo AWS account

L'`list-certificates` esempio seguente elenca tutti i certificati registrati nel tuo account. Se hai più del limite di paging predefinito di 25, puoi utilizzare il valore di `nextMarker` risposta di questo comando e fornirlo al comando successivo per ottenere il successivo batch di risultati. Ripetere l'operazione fino a quando non viene `nextMarker` restituito un valore.

```
aws iot list-certificates
```

Output:

```
{
  "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
    }
  ]
}

```

- Per API i dettagli, vedere [ListCertificates](#) in AWS CLI Command Reference.

## list-custom-metrics

Il seguente esempio di codice mostra come utilizzare `list-custom-metrics`.

### AWS CLI

Per elencare le metriche personalizzate

L'`list-custom-metrics` esempio seguente elenca tutte le metriche personalizzate.

```

aws iot list-custom-metrics \
  --region us-east-1

```

Output:

```
{
  "metricNames": [
    "batteryPercentage"
  ]
}
```

Per ulteriori informazioni, consulta [Metriche personalizzate](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [ListCustomMetrics AWS CLI Command Reference](#).

## list-dimensions

Il seguente esempio di codice mostra come utilizzare `list-dimensions`.

AWS CLI

Per elencare le dimensioni del tuo AWS account

L'`list-dimensions` esempio seguente elenca tutte le dimensioni di AWS IoT Device Defender definite nell'account AWS .

```
aws iot list-dimensions
```

Output:

```
{
  "dimensionNames": [
    "TopicFilterForAuthMessages",
    "TopicFilterForActivityMessages"
  ]
}
```

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ListDimensions](#) in AWS CLI Command Reference.

## list-domain-configurations

Il seguente esempio di codice mostra come utilizzare `list-domain-configurations`.

## AWS CLI

Per elencare le configurazioni dei domini

L'`list-domain-configurations` seguente elenca le configurazioni di dominio AWS dell'account con il tipo di servizio specificato.

```
aws iot list-domain-configurations \  
  --service-type "DATA"
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```



Per ulteriori informazioni, consulta [Configurable Endpoints](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListDomainConfigurations](#) Reference.

## list-indices

Il seguente esempio di codice mostra come utilizzare `list-indices`.

### AWS CLI

Per elencare gli indici di ricerca configurati

L'`list-indices` esempio seguente elenca tutti gli indici di ricerca configurati nel tuo AWS account. Se non hai abilitato l'indicizzazione degli oggetti, potresti non avere alcun indice.

```
aws iot list-indices
```

Output:

```
{
  "indexNames": [
    "AWS_Things"
  ]
}
```

Per ulteriori informazioni, consulta [Managing Thing Indexing](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ListIndices](#) in AWS CLI Command Reference.

## list-job-executions-for-job

Il seguente esempio di codice mostra come utilizzare `list-job-executions-for-job`.

### AWS CLI

Per elencare le offerte di lavoro presenti nel tuo AWS account

L'`list-job-executions-for-job` esempio seguente elenca tutte le esecuzioni di job relative a un job nel tuo AWS account, specificato da `jobId`

```
aws iot list-job-executions-for-job \
  --job-id my-ota-job
```

**Output:**

```
{
  "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
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta [Creating and Managing Jobs \(CLI\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [ListJobExecutionsForJob AWS CLI Command Reference](#).

**list-job-executions-for-thing**

Il seguente esempio di codice mostra come utilizzare `list-job-executions-for-thing`.

**AWS CLI**

Per elencare i lavori che sono stati eseguiti per una cosa

L'`list-job-executions-for-things`esempio seguente elenca tutti i lavori che sono stati eseguiti per l'oggetto denominato `MyRaspberryPi`.

```
aws iot list-job-executions-for-thing \
  --thing-name "MyRaspberryPi"
```

**Output:**

```
{
  "executionSummaries": [
    {
```

```
    "jobId": "example-job-01",
    "jobExecutionSummary": {
      "status": "QUEUED",
      "queuedAt": 1560787023.636,
      "lastUpdatedAt": 1560787023.636,
      "executionNumber": 1
    }
  ]
}
```

Per ulteriori informazioni, consulta [Creating and Managing Jobs \(CLI\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [ListJobExecutionsForThing AWS CLI Command Reference](#).

## list-jobs

Il seguente esempio di codice mostra come utilizzare `list-jobs`.

### AWS CLI

Per elencare le offerte di lavoro presenti nel tuo AWS account

L'`list-jobs` esempio seguente elenca tutti i lavori del tuo AWS account, ordinati in base allo stato del lavoro.

```
aws iot list-jobs
```

Output:

```
{
  "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
    }
  ]
}
```

```
}
```

Per ulteriori informazioni, consulta [Creating and Managing Jobs \(CLI\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [ListJobs AWS CLI Command Reference](#).

## list-mitigation-actions

Il seguente esempio di codice mostra come utilizzare `list-mitigation-actions`.

### AWS CLI

Per elencare tutte le azioni di mitigazione definite

L'`list-mitigation-actions` esempio seguente elenca tutte le azioni di mitigazione definite per l' AWS account e la regione. Per ogni azione sono elencati il nome ARN e la data di creazione.

```
aws iot list-mitigation-actions
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [ListMitigationActions \(Mitigation Action Commands\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ListMitigationActions](#) in AWS CLI Command Reference.

## list-mitigations-actions

Il seguente esempio di codice mostra come utilizzare `list-mitigations-actions`.

### AWS CLI

Per elencare tutte le azioni di mitigazione definite

L'`list-mitigations-actions`esempio seguente elenca tutte le azioni di mitigazione definite per l' AWS account e la regione. Per ogni azione sono elencati il nome ARN e la data di creazione.

```
aws iot list-mitigation-actions
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [ListMitigationActions \(Mitigation Action Commands\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ListMitigationsActions](#) in AWS CLI Command Reference.

## list-ota-updates

Il seguente esempio di codice mostra come utilizzare `list-ota-updates`.

### AWS CLI

Per elencare OTA gli aggiornamenti relativi all'account

L'`list-ota-updates` esempio seguente elenca gli OTA aggiornamenti disponibili.

```
aws iot list-ota-updates
```

Output:

```
{
  "otaUpdates": [
    {
      "otaUpdateId": "itsaupdate",
      "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/itsaupdate",
      "creationDate": 1557863215.995
    }
  ]
}
```

Per ulteriori informazioni, vedere [ListOTAUpdates](#) in AWS IoT API Reference.

- Per API i dettagli, vedere [ListOtaUpdates](#) in AWS CLI Command Reference.

## list-outgoing-certificates

Il seguente esempio di codice mostra come utilizzare `list-outgoing-certificates`.

### AWS CLI

Per elencare i certificati trasferiti su un altro AWS account

L'`list-outgoing-certificates` esempio seguente elenca tutti i certificati dei dispositivi che stanno per essere trasferiti su un altro AWS account utilizzando il `transfer-certificate` comando.

```
aws iot list-outgoing-certificates
```

Output:

```
{
  "outgoingCertificates": [
    {
      "certificateArn": "arn:aws:iot:us-west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "certificateId":
      "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
      "transferredTo": "030714055129",
      "transferDate": 1569427780.441,
      "creationDate": 1569363250.557
    }
  ]
}
```

Per ulteriori informazioni, [ListOutgoingCertificates](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [ListOutgoingCertificates](#) in AWS CLI Command Reference.

## list-policies

Il seguente esempio di codice mostra come utilizzare `list-policies`.

AWS CLI

Per elencare le politiche definite nel tuo AWS account

L'`list-policies` esempio seguente elenca tutte le politiche definite nel tuo AWS account.

```
aws iot list-policies
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [AWS IoT Policies](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [ListPolicies](#) in AWS CLI Command Reference.

## list-policy-versions

Il seguente esempio di codice mostra come utilizzare `list-policy-versions`.

### AWS CLI

Esempio 1: per visualizzare tutte le versioni di una politica

L'`list-policy-versions` esempio seguente elenca tutte le versioni della politica specificata e le relative date di creazione.

```
aws iot list-policy-versions \
  --policy-name LightBulbPolicy
```

Output:

```
{
  "policyVersions": [
    {
      "versionId": "2",
      "isDefaultVersion": true,
      "createDate": 1559925941.924
    },
    {
      "versionId": "1",
```

```
        "isDefaultVersion": false,  
        "createDate": 1559925941.924  
    }  
]  
}
```

Per ulteriori informazioni, consulta [AWS IoT Policies](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [ListPolicyVersions](#) in AWS CLI Command Reference.

## list-principal-things

Il seguente esempio di codice mostra come utilizzare `list-principal-things`.

### AWS CLI

Per elencare le cose allegate a un preside

L'`list-principal-things` esempio seguente elenca gli elementi collegati al principale specificato da un ARN.

```
aws iot list-principal-things \  
  --principal arn:aws:iot:us-  
west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a35159767055278e8
```

Output:

```
{  
  "things": [  
    "DeskLamp",  
    "TableLamp"  
  ]  
}
```

Per ulteriori informazioni, [ListPrincipalThings](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [ListPrincipalThings](#) in AWS CLI Command Reference.

## list-provisioning-template-versions

Il seguente esempio di codice mostra come utilizzare `list-provisioning-template-versions`.

## AWS CLI

Per elencare le versioni dei modelli di provisioning

L'`list-provisioning-template-versions` seguente elenca le versioni disponibili del modello di provisioning specificato.

```
aws iot list-provisioning-template-versions \  
  --template-name "widget-template"
```

Output:

```
{  
  "versions": [  
    {  
      "versionId": 1,  
      "creationDate": 1574800471.339,  
      "isDefaultVersion": true  
    },  
    {  
      "versionId": 2,  
      "creationDate": 1574801192.317,  
      "isDefaultVersion": false  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [AWS IoT Secure Tunneling](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListProvisioningTemplateVersions](#) Reference.

## **list-provisioning-templates**

Il seguente esempio di codice mostra come utilizzare `list-provisioning-templates`.

### AWS CLI

Per elencare i modelli di provisioning

L'`list-provisioning-templates` seguente elenca tutti i modelli di provisioning presenti nel tuo AWS account.

```
aws iot list-provisioning-templates
```

Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta [AWS IoT Secure Tunneling](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListProvisioningTemplates](#) Reference.

## list-role-aliases

Il seguente esempio di codice mostra come utilizzare `list-role-aliases`.

AWS CLI

Per elencare gli alias dei ruoli AWS IoT nel tuo account AWS

L'`list-role-aliases` esempio seguente elenca gli alias dei ruoli AWS IoT nel tuo AWS account.

```
aws iot list-role-aliases
```

Output:

```
{
  "roleAliases": [
    "ResidentAlias",
```

```
    "ElectricianAlias"  
  ]  
}
```

Per ulteriori informazioni, [ListRoleAliases](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [ListRoleAliases](#) in AWS CLI Command Reference.

## list-scheduled-audits

Il seguente esempio di codice mostra come utilizzare `list-scheduled-audits`.

### AWS CLI

Per elencare gli audit programmati per il tuo account AWS

L'`list-scheduled-audits` esempio seguente elenca tutti gli audit programmati per il tuo AWS account.

```
aws iot list-scheduled-audits
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Audit Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ListScheduledAudits](#) in AWS CLI Command Reference.

## list-security-profiles-for-target

Il seguente esempio di codice mostra come utilizzare `list-security-profiles-for-target`.

### AWS CLI

Per elencare i profili di sicurezza collegati a una destinazione

L'`list-security-profiles-for-target` esempio seguente elenca i profili di sicurezza AWS IoT Device Defender collegati a dispositivi non registrati.

```
aws iot list-security-profiles-for-target \
  --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/
  unregistered-things"
```

Output:

```
{
  "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"
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ListSecurityProfilesForTarget](#) in AWS CLI Command Reference.

## list-security-profiles

Il seguente esempio di codice mostra come utilizzare `list-security-profiles`.

## AWS CLI

Per elencare i profili di sicurezza del tuo AWS account

L'`list-security-profiles` seguente elenca tutti i profili di sicurezza AWS IoT Device Defender definiti nell'account AWS .

```
aws iot list-security-profiles
```

Output:

```
{
  "securityProfileIdentifiers": [
    {
      "name": "Testprofile",
      "arn": "arn:aws:iot:us-west-2:123456789012:securityprofile/Testprofile"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ListSecurityProfiles](#) in AWS CLI Command Reference.

## `list-streams`

Il seguente esempio di codice mostra come utilizzare `list-streams`.

## AWS CLI

Per elencare gli stream presenti nell'account

L'`list-streams` seguente elenca tutti gli stream del tuo AWS account.

```
aws iot list-streams
```

Output:

```
{
  "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."
  }
]
}

```

Per ulteriori informazioni, [ListStreams](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [ListStreams](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per visualizzare i tag e i relativi valori associati a una risorsa

L'`list-tags-for-resource` esempio seguente visualizza i tag e i valori associati al gruppo di oggetti `LightBulbs`.

```

aws iot list-tags-for-resource \
  --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"

```

Output:

```

{
  "tags": [
    {
      "Key": "Assembly",
      "Value": "Fact1NW"
    },
    {
      "Key": "MyTag",

```



```
        "Value": "777"
      }
    ]
  }
```

Per ulteriori informazioni, consulta [Tagging Your AWS IoT Resources](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## list-targets-for-policy

Il seguente esempio di codice mostra come utilizzare `list-targets-for-policy`.

### AWS CLI

Per elencare i principi associati a una policy AWS IoT

L'`list-targets-for-policy` esempio seguente elenca i certificati dei dispositivi a cui è allegata la policy specificata.

```
aws iot list-targets-for-policy \
  --policy-name UpdateDeviceCertPolicy
```

Output:

```
{
  "targets": [
    "arn:aws:iot:us-west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
    "arn:aws:iot:us-west-2:123456789012:cert/d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be"
  ]
}
```

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [ListTargetsForPolicy](#) in AWS CLI Command Reference.

## list-targets-for-security-profile

Il seguente esempio di codice mostra come utilizzare `list-targets-for-security-profile`.

## AWS CLI

Per elencare le destinazioni a cui viene applicato un profilo di sicurezza

L'`list-targets-for-security-profile` esempio seguente elenca le destinazioni a cui viene applicato il profilo di sicurezza AWS IoT Device Defender denominato `PossibleIssue`.

```
aws iot list-targets-for-security-profile \  
  --security-profile-name Testprofile
```

Output:

```
{  
  "securityProfileTargets": [  
    {  
      "arn": "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"  
    },  
    {  
      "arn": "arn:aws:iot:us-west-2:123456789012:all/registered-things"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ListTargetsForSecurityProfile](#) in AWS CLI Command Reference.

## `list-thing-groups-for-thing`

Il seguente esempio di codice mostra come utilizzare `list-thing-groups-for-thing`.

### AWS CLI

Per elencare i gruppi a cui appartiene un oggetto

L'`list-thing-groups-for-thing` esempio seguente elenca i gruppi a cui appartiene l'oggetto specificato.

```
aws iot list-thing-groups-for-thing \  
  --thing-name MyLightBulb
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [ListThingGroupsForThing](#) in AWS CLI Command Reference.

## list-thing-groups

Il seguente esempio di codice mostra come utilizzare `list-thing-groups`.

### AWS CLI

Per elencare i gruppi di cose definiti nel tuo AWS account

L'esempio seguente elenca tutti i gruppi di cose definiti nell'AWS account.

```
aws iot list-thing-groups
```

Output:

```
{
  "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"
    }
  ]
}
```

```
]
}
```

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [ListThingGroups](#) in AWS CLI Command Reference.

## **list-thing-principals**

Il seguente esempio di codice mostra come utilizzare `list-thing-principals`.

### AWS CLI

Per elencare i principi associati a una cosa

L'`list-thing-principals` esempio seguente elenca i principali (certificati X.509, IAM utenti, gruppi, ruoli, identità Amazon Cognito o identità federate) associati all'oggetto specificato.

```
aws iot list-thing-principals \  
  --thing-name MyRaspberryPi
```

Output:

```
{  
  "principals": [  
    "arn:aws:iot:us-  
west-2:123456789012:cert/33475ac865079a5ffd5ecd44240640349293facc760642d7d8d5dbb6b4c86893"  
  ]  
}
```

Per ulteriori informazioni, [ListThingPrincipals](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [ListThingPrincipals](#) in AWS CLI Command Reference.

## **list-thing-types**

Il seguente esempio di codice mostra come utilizzare `list-thing-types`.

### AWS CLI

Per elencare i tipi di oggetti definiti

L'`list-thing-types` seguente visualizza un elenco di tipi di oggetti definiti nell' AWS account.

```
aws iot list-thing-types
```

Output:

```
{
  "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
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta [Thing Types](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [ListThingTypes](#) in AWS CLI Command Reference.

## **list-things-in-billing-group**

Il seguente esempio di codice mostra come utilizzare `list-things-in-billing-group`.

AWS CLI

Per elencare gli elementi in un gruppo di fatturazione

L'`list-things-in-billing-group` seguente elenca gli elementi che si trovano nel gruppo di fatturazione specificato.

```
aws iot list-things-in-billing-group \  
  --billing-group-name GroupOne
```

Output:

```
{  
  "things": [  
    "MyOtherLightBulb",  
    "MyLightBulb"  
  ]  
}
```

Per ulteriori informazioni, consulta [Billing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [ListThingsInBillingGroup AWS CLI Command Reference](#).

## list-things-in-thing-group

Il seguente esempio di codice mostra come utilizzare `list-things-in-thing-group`.

AWS CLI

Per elencare gli elementi che appartengono a un gruppo

L'`list-things-in-thing-group` esempio seguente elenca gli oggetti che appartengono al gruppo di oggetti specificato.

```
aws iot list-things-in-thing-group \  
  --thing-group-name LightBulbs
```

Output:

```
{  
  "things": [  
    "MyLightBulb"  
  ]  
}
```

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [ListThingsInThingGroup](#) in AWS CLI Command Reference.

## list-things

Il seguente esempio di codice mostra come utilizzare `list-things`.

### AWS CLI

Esempio 1: Per elencare tutte le cose nel registro

L'`list-things` seguente elenca gli oggetti (dispositivi) definiti nel registro AWS IoT per il tuo AWS account.

```
aws iot list-things
```

Output:

```
{
  "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
    }
  ]
}
```

Esempio 2: elencare gli elementi definiti che hanno un attributo specifico

L'`list-things` seguente visualizza un elenco di cose che hanno un attributo denominato `wattage`.

```
aws iot list-things \
  --attribute-name wattage
```

Output:

```
{
  "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
      }
    ]
  }
```

Per ulteriori informazioni, consulta [How to Manage Things with the Registry](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [ListThings](#) in AWS CLI Command Reference.

## list-topic-rule-destinations

Il seguente esempio di codice mostra come utilizzare `list-topic-rule-destinations`.

AWS CLI

Per elencare le destinazioni delle regole tematiche

L'`list-topic-rule-destination`sempio seguente elenca tutte le destinazioni delle regole tematiche che avete definito nella AWS regione corrente.

```
aws iot list-topic-rule-destinations
```

Output:

```
{
  "destinationSummaries": [
    {
      "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "status": "ENABLED",
      "httpUrlSummary": {
        "confirmationUrl": "https://example.com"
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta [Lavorare con le destinazioni delle regole tematiche](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [ListTopicRuleDestinations AWS CLI](#) Command Reference.

## list-topic-rules

Il seguente esempio di codice mostra come utilizzare `list-topic-rules`.

### AWS CLI

Per elencare le tue regole

L'`list-topic-rules` esempio seguente elenca tutte le regole che avete definito.

```
aws iot list-topic-rules
```

Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta [Viewing Your Rules](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [ListTopicRules](#) in AWS CLI Command Reference.

## list-v2-logging-levels

Il seguente esempio di codice mostra come utilizzare `list-v2-logging-levels`.

## AWS CLI

Per elencare i livelli di registrazione

L'`list-v2-logging-levels` seguente elenca i livelli di registrazione configurati. Se i livelli di registrazione non sono stati impostati, `NotConfiguredException` si verifica a quando si esegue questo comando.

```
aws iot list-v2-logging-levels
```

Output:

```
{
  "logTargetConfigurations": [
    {
      "logTarget": {
        "targetType": "DEFAULT"
      },
      "logLevel": "ERROR"
    }
  ]
}
```

- Per API i dettagli, vedere [ListV2 LoggingLevels](#) in AWS CLI Command Reference.

## `list-violation-events`

Il seguente esempio di codice mostra come utilizzare `list-violation-events`

## AWS CLI

Per elencare le violazioni del profilo di sicurezza durante un periodo di tempo

L'`list-violation-events` seguente elenca le violazioni che si sono verificate tra il 5 giugno 2019 e il 12 giugno 2019 per tutti i profili di sicurezza AWS IoT Device Defender per l'AWS account corrente e AWS la regione.

```
aws iot list-violation-events \
  --start-time 1559747125 \
  --end-time 1560351925
```

## Output:

```
{
  "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": "c8a9466a093d3b7b35cd44ca58bdbeab",
      "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
  }
]
}
```

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ListViolationEvents](#) in AWS CLI Command Reference.

## register-ca-certificate

Il seguente esempio di codice mostra come utilizzare `register-ca-certificate`.

### AWS CLI

Per registrare un certificato di autorità di certificazione (CA)

L'esempio seguente registra un certificato CA. Il comando fornisce il certificato CA e un certificato di verifica della chiave che dimostra che l'utente possiede la chiave privata associata al certificato CA.

```
aws iot register-ca-certificate \
  --ca-certificate file://rootCA.pem \
  --verification-cert file://verificationCert.pem
```

Output:

```
{
  "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467",
  "certificateId":
"f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467"
}
```

Per ulteriori informazioni, consulta [RegisterCACertificate](#) in AWS IoT API Reference.

- Per API i dettagli, vedere [RegisterCaCertificate](#) in AWS CLI Command Reference.

## register-certificate

Il seguente esempio di codice mostra come utilizzare `register-certificate`.

### AWS CLI

Per registrare un certificato di dispositivo autofirmato

L'esempio seguente registra il certificato del dispositivo `deviceCert.pem` firmato dal certificato `rootCA.pem` CA. Il certificato CA deve essere registrato prima di essere utilizzato per registrare un certificato di dispositivo autofirmato. Il certificato autofirmato deve essere firmato con lo stesso certificato CA passato a questo comando.

```
aws iot register-certificate \  
  --certificate-pem file://deviceCert.pem \  
  --ca-certificate-pem file://rootCA.pem
```

Output:

```
{  
  "certificateArn": "arn:aws:iot:us-  
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",  
  "certificateId":  
  "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"  
}
```

Per ulteriori informazioni, [RegisterCertificate](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [RegisterCertificate](#) in AWS CLI Command Reference.

## register-thing

Il seguente esempio di codice mostra come utilizzare `register-thing`.

### AWS CLI

Per registrare una cosa

L'esempio seguente registra un oggetto utilizzando un modello di provisioning.

```
aws iot register-thing \  
  --thing-type type
```

```

--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

```

## Output:

```

{
  "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAKGgAwIBAgIUYLk81I35cIppobpw
Hi0J2jNjboIwDQYJKoZIhvcNAQEL
\nBQAwTTFLEkGA1UECwxCQW1hem9uIFd1YiBTZXJ2aWNlcyBPPUFtYXpvbi
5jb20g\nSW5jLiBMPVNlYXR0bGUgU1Q9V2FzaGluZ3RvbiBDPVVtMB4XDTEwMDcyMzE2NDUw
\n0VoXDTQ5MTIzMT
IzNTk1OVowHjEcMBoGA1UEAwwTQVdTIElvcyBDbDZXJ0aWZpY2F0\nZTCCASIUwDQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBA071uADhdBajqTmgrpMV5\nmCFfBZQRMo1MdtVoZr2X+M4MzL
+RARrtUzH9a2SMAckeX8Keb1I0TKzORI
RDXnyE
\n61V0wjgAsd0ku22rFxex4eG2ikha7pYYkvuToqA7L3TxItRvfKrxRI4ZfJoFPip4\nnKqiuBJVNOGKTCq
Hd1RN0rddwwu6kFJLeKDMEXAMPLEdUF0N+qfR9yKnZQkm
+g6Q2\nnGXu7u0W3hn6n1RN8qVoka0uW12p53xM7oHVz
Gf+cxKBx1b0hGkp6yCfTskUBm3Sp\nn9zLw35kiHXVm4EVpwnlNk6XcIGIkw8a/iy4pzmvuGAANY1/uU/
zgCjymw
ZT5S30\nnBV0CAwEAAANgMF4wHwYDVR0jBBgwFoAUGx0tCcU3q2n1WXAuUCv6hugXjKswHQYD
\nVR00BBYEF0VtvZ
9Aj2RYFnkX7Iu01XTRUdxgMAwGA1UdEwEB/wQCMAAwDgYDVR0P\nnAQH/
BAQDAgeAMA0GCSqGSIb3DQEBCwUAA4IB
AQXCQCcp0tubS5ft0sDMTcP/jNX
\nDHyaRxmjpSc2aCdmm7WX591TKWyAdxGAVqaDVWqTo0oXI7tZ8w7aINlGi5
pXnifx\nn3SBebMUoBbTktrC97yUaeL025mCFv8emDnTR/fe7PTsBKjW0g/rrfpwBxZLXDFwN
\nnqkQjy3EDfifj2
6j0xYIqqWMPogyn4sr0CKynS5wMJuQZ1HQ0nabVwnwK4Y0Mflp
\np9+4susFUR9aT3BT1AcIwqSpzh1Khh4Iz7ND
kRn4amsUT210jg/z0010w+BTHcVQ\nnJly8XDu0CWSu04q6SnaBzHmlySIajxuRTP/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"
}
```

Per ulteriori informazioni, consulta la sezione [Provisioning by trusted user](#) nella AWS IoT Core Developers Guide.

- Per API i dettagli, vedere [RegisterThing](#) in AWS CLI Command Reference.

## reject-certificate-transfer

Il seguente esempio di codice mostra come utilizzare `reject-certificate-transfer`.

### AWS CLI

Per rifiutare il trasferimento di un certificato

L'esempio seguente rifiuta il trasferimento del certificato del dispositivo specificato da un altro AWS account.

```
aws iot reject-certificate-transfer \
  --certificate-
  id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78addd5e605d630e05c7fc8
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Trasferire un certificato su un altro account](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [RejectCertificateTransfer AWS CLI](#) Command Reference.

## remove-thing-from-billing-group

Il seguente esempio di codice mostra come utilizzare `remove-thing-from-billing-group`.

### AWS CLI

Per rimuovere un elemento da un gruppo di fatturazione

L'`remove-thing-from-billing-group` seguente rimuove l'elemento specificato da un gruppo di fatturazione.

```
aws iot remove-thing-from-billing-group \  
  --billing-group-name GroupOne \  
  --thing-name MyOtherLightBulb
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Billing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [RemoveThingFromBillingGroup AWS CLI Command Reference](#).

## **remove-thing-from-thing-group**

Il seguente esempio di codice mostra come utilizzare `remove-thing-from-thing-group`.

### AWS CLI

Per rimuovere un elemento da un gruppo di oggetti

L'`remove-thing-from-thing-group` seguente rimuove l'oggetto specificato da un gruppo di oggetti.

```
aws iot remove-thing-from-thing-group \  
  --thing-name bulb7 \  
  --thing-group-name DeadBulbs
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere Thing Groups < <https://docs.aws.amazon.com/iot/latest/developerguide/thing-groups.html> > nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [RemoveThingFromThingGroup](#) in AWS CLI Command Reference.

## **replace-topic-rule**

Il seguente esempio di codice mostra come utilizzare `replace-topic-rule`.

### AWS CLI

Per aggiornare la definizione delle regole di un argomento

L'replace-topic-rule esempio seguente aggiorna la regola specificata per inviare un SNS avviso quando i valori del livello di umidità del suolo sono troppo bassi.

```
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\"}"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creating an AWS IoT Rule](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [ReplaceTopicRule AWS CLI Command Reference](#).

## search-index

Il seguente esempio di codice mostra come utilizzare search-index.

### AWS CLI

Per interrogare l'indice dell'oggetto

L'search-index esempio seguente interroga l'AWS\_Things indice per cercare elementi che hanno un tipo di LightBulb.

```
aws iot search-index \
  --index-name "AWS_Things" \
  --query-string "thingTypeName:LightBulb"
```

Output:

```
{
  "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
    }
  }
]
}
```

Per ulteriori informazioni, consulta [Managing Thing Indexing](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [SearchIndex](#) in AWS CLI Command Reference.

## set-default-authorizer

Il seguente esempio di codice mostra come utilizzare `set-default-authorizer`.

### AWS CLI

Per impostare un autorizzatore predefinito

L'`set-default-authorizer` seguente imposta l'autorizzatore personalizzato `CustomAuthorizer` denominato autorizzatore predefinito.

```
aws iot set-default-authorizer \  
  --authorizer-name CustomAuthorizer
```

Output:

```
{  
  "authorizerName": "CustomAuthorizer",  
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/  
CustomAuthorizer"  
}
```

Per ulteriori informazioni, [CreateDefaultAuthorizer](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [SetDefaultAuthorizer](#) in AWS CLI Command Reference.

## set-default-policy-version

Il seguente esempio di codice mostra come utilizzare `set-default-policy-version`.

### AWS CLI

Per impostare la versione predefinita di una politica

L'`set-default-policy-version` seguente imposta la versione predefinita 2 per la politica denominata `UpdateDeviceCertPolicy`.

```
aws iot set-default-policy-version \  
  --policy-name UpdateDeviceCertPolicy \  
  --policy-version-id 2
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [SetDefaultPolicyVersion](#) in AWS CLI Command Reference.

## set-v2-logging-level

Il seguente esempio di codice mostra come utilizzare `set-v2-logging-level`.

### AWS CLI

Per impostare il livello di registrazione per un gruppo di oggetti

L'`set-v2-logging-level` esempio seguente imposta il livello di registrazione per registrare gli avvisi per il gruppo di oggetti specificato.

```
aws iot set-v2-logging-level \  
  --log-target "{\"targetType\":\"THING_GROUP\",\"targetName\":\"LightBulbs\"}" \  
  --log-level WARN
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [SetV2 LoggingLevel](#) in Command Reference.AWS CLI

## set-v2-logging-options

Il seguente esempio di codice mostra come utilizzare `set-v2-logging-options`

### AWS CLI

Per impostare le opzioni di registrazione

L'`set-v2-logging-options` esempio seguente imposta il livello di dettaglio della registrazione predefinito su `ERROR` e specifica quello da utilizzare per la ARN registrazione.

```
aws iot set-v2-logging-options \  
  --default-log-level ERROR \  
  --role-arn "arn:aws:iam::094249569039:role/service-role/iotLoggingRole"
```

Questo comando non produce alcun output.

- [Per i API dettagli, vedere SetV2 in Command Reference. LoggingOptions AWS CLI](#)

## start-audit-mitigation-actions-task

Il seguente esempio di codice mostra come utilizzare `start-audit-mitigation-actions-task`

### AWS CLI

Per applicare un'azione di mitigazione ai risultati di un audit

L'esempio seguente applica l'azione `ResetPolicyVersionAction` (che cancella la politica) al singolo risultato specificato.

```
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"
```

Output:

```
{
  "taskId": "myActionsTaskId"
}
```

Per ulteriori informazioni, consulta [StartAuditMitigationActionsTask \(Mitigation Action Commands\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [StartAuditMitigationActionsTask](#) in AWS CLI Command Reference.

## start-on-demand-audit-task

Il seguente esempio di codice mostra come utilizzare `start-on-demand-audit-task`.

### AWS CLI

Per avviare subito un audit

L'esempio seguente avvia un audit AWS IoT Device Defender ed esegue tre controlli dei certificati.

```
aws iot start-on-demand-audit-task \
```

```
--target-check-
```

```
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK REVOKED_CA_CERTIFICATE
```

Output:

```
{
  "taskId": "a3aea009955e501a31b764abe1bebd3d"
}
```

Per ulteriori informazioni, consulta [Audit Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [StartOnDemandAuditTask](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

AWS CLI

Per specificare una chiave e un valore di tag per una risorsa

L'esempio seguente applica il tag con una chiave `Assembly` e il valore `Fact1NW` al gruppo di cose `LightBulbs`.

```
aws iot tag-resource \
  --tags Key=Assembly,Value="Fact1NW" \
  --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Your AWS IoT Resources](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [TagResource AWS CLI Command Reference](#).

## test-authorization

Il seguente esempio di codice mostra come utilizzare `test-authorization`.

AWS CLI

Per testare le tue politiche AWS IoT



L'operazione `test-authorization` seguente verifica le politiche AWS IoT associate al principale specificato.

```
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
```

Output:

```
{  
  "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": []
}
]
}

```

Per ulteriori informazioni, [TestAuthorization](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [TestAuthorization](#) in AWS CLI Command Reference.

## test-invoke-authorizer

Il seguente esempio di codice mostra come utilizzare `test-invoke-authorizer`.

### AWS CLI

Per testare il tuo autorizzatore personalizzato

L'`test-invoke-authorizer` esempio seguente verifica il tuo autorizzatore personalizzato.

```

aws iot test-invoke-authorizer \
  --authorizer-name IoTAuthorizer \
  --token allow \
  --token-signature "mE0GvaHqy9nER/
FdgtJX5LXYEJ3b3vE7t1gEszc0TKGgLKWXTnPkb2AbKn0AZ8lGyoN5dVtWDWVmr25m7+
+zjbYIMk2TBvyGXh0mvKFBPkdgyA43KL6SiZy0cTqLPMcQDsP7VX2rXr7CTowCxSNKphGXdq0/
I5dQ+J06KUaHwCmupt0/MejKtaNwiiA064j6wpr0AUwG5S1IYFuRd0X
+wfo8pb0DubAIX1Ua705kuhRUcTx4SxUSHEYKmN4IDEvLB6FsIr0B2wvB7y4iPmcajxzG102ExvyCUNctCV9dYLRGJj"

```

Output:

```

{
  "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
}

```

Per ulteriori informazioni, [TestInvokeAuthorizer](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [TestInvokeAuthorizer](#) in AWS CLI Command Reference.

## transfer-certificate

Il seguente esempio di codice mostra come utilizzare `transfer-certificate`.

### AWS CLI

Per trasferire il certificato di un dispositivo su un altro AWS account

L'`transfer-certificate` esempio seguente trasferisce il certificato di un dispositivo su un altro AWS account. Il certificato e l' AWS account sono identificati da un ID.

```

aws iot transfer-certificate \
  --certificate-
id 488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142 \
  --target-aws-account 030714055129

```

Output:

```

{
  "transferredCertificateArn": "arn:aws:iot:us-
west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"
}

```

Per ulteriori informazioni, consulta [Trasferire un certificato su un altro account](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [TransferCertificate AWS CLI Command Reference](#).

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere una chiave di tag da una risorsa

L'esempio seguente rimuove il tag `MyTag` e il relativo valore dal gruppo di cose `LightBulbs`.

```
command
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Your AWS IoT Resources](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [UntagResource AWS CLI Command Reference](#).

## update-account-audit-configuration

Il seguente esempio di codice mostra come utilizzare `update-account-audit-configuration`.

### AWS CLI

Esempio 1: abilitare le notifiche di Amazon per SNS le notifiche di controllo

L'esempio seguente abilita SNS le notifiche di Amazon notification for AWS IoT Device Defender per le notifiche di audit, specificando una destinazione e il ruolo utilizzato per scrivere su tale destinazione.

```
aws iot update-account-audit-configuration \
  --audit-notification-target-configurations "SNS={targetArn=\"arn:aws:sns:us-
west-2:123456789012:ddauidits\",roleArn=\"arn:aws:iam::123456789012:role/service-
role/AWSIoTDeviceDefenderAudit\",enabled=true}"
```

Questo comando non produce alcun output.

Esempio 2: per abilitare un controllo di controllo

L'update-account-audit-configurationesempio seguente abilita il controllo di controllo AWS IoT Device Defender denominato. AUTHENTICATED\_COGNITO\_ROLE\_OVERLY\_PERMISSIVE\_CHECK Non è possibile disabilitare un controllo di controllo se fa parte targetCheckNames di uno o più audit programmati per l' AWS account.

```
aws iot update-account-audit-configuration \  
  --audit-check-configurations \  
  "{\"AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK\":{\"enabled\":true}}"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Audit Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [UpdateAccountAuditConfiguration](#) in AWS CLI Command Reference.

## update-audit-suppression

Il seguente esempio di codice mostra come utilizzareupdate-audit-suppression.

AWS CLI

Per aggiornare un audit, la soppressione dei risultati

L'update-audit-suppressionesempio seguente aggiorna la data di scadenza della soppressione di un risultato di controllo al 21/09/2020.

```
aws iot update-audit-suppression \  
  --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \  
  --resource-identifier deviceCertificateId=c7691e<shortened> \  
  --no-suppress-indefinitely \  
  --expiration-date 2020-09-21
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Audit finding suppressions](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateAuditSuppression](#)Reference.

## update-authorizer

Il seguente esempio di codice mostra come utilizzareupdate-authorizer.

## AWS CLI

Per aggiornare un autorizzatore personalizzato

L'update-authorizer esempio seguente: lo stato CustomAuthorizer2 di. INACTIVE

```
aws iot update-authorizer \  
  --authorizer-name CustomAuthorizer2 \  
  --status INACTIVE
```

Output:

```
{  
  "authorizerName": "CustomAuthorizer2",  
  "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/  
CustomAuthorizer2"  
}
```

Per ulteriori informazioni, [UpdateAuthorizer](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [UpdateAuthorizer](#) in AWS CLI Command Reference.

## update-billing-group

Il seguente esempio di codice mostra come utilizzare update-billing-group.

## AWS CLI

Per aggiornare le informazioni su un gruppo di fatturazione

L'update-billing-group esempio seguente aggiorna la descrizione per il gruppo di fatturazione specificato.

```
aws iot update-billing-group \  
  --billing-group-name GroupOne \  
  --billing-group-properties "billingGroupDescription=\"Primary bulb billing group  
\""
```

Output:

```
{
```

```
"version": 2
}
```

Per ulteriori informazioni, consulta [Billing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [UpdateBillingGroup AWS CLI Command Reference](#).

## update-ca-certificate

Il seguente esempio di codice mostra come utilizzare `update-ca-certificate`.

### AWS CLI

Per aggiornare un certificato di autorità di certificazione (CA)

L'esempio seguente imposta lo `ACTIVE` status del certificato CA specificato.

```
aws iot update-ca-certificate \
  --certificate-
  id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467 \
  --new-status ACTIVE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [UpdateCACertificate](#) in AWS IoT API Reference.

- Per API i dettagli, vedere [UpdateCaCertificate](#) in AWS CLI Command Reference.

## update-certificate

Il seguente esempio di codice mostra come utilizzare `update-certificate`.

### AWS CLI

Per aggiornare il certificato di un dispositivo

L'esempio seguente imposta lo `INACTIVE` status del certificato del dispositivo specificato.

```
aws iot update-certificate \
```

```
--certificate-  
id d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be \  
--new-status INACTIVE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [UpdateCertificate](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [UpdateCertificate](#) in AWS CLI Command Reference.

## update-custom-metric

Il seguente esempio di codice mostra come utilizzare `update-custom-metric`.

### AWS CLI

Per aggiornare una metrica personalizzata

L'`update-custom-metric` esempio seguente aggiorna una metrica personalizzata con una nuova `display-name`

```
aws iot update-custom-metric \  
--metric-name batteryPercentage \  
--display-name 'remaining battery percentage on device' \  
--region us-east-1
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Metriche personalizzate](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [UpdateCustomMetric AWS CLI](#) Command Reference.



## update-dimension

Il seguente esempio di codice mostra come utilizzare `update-dimension`.

### AWS CLI

Per aggiornare una dimensione

L'esempio seguente aggiorna una dimensione.

```
aws iot update-dimension \  
  --name TopicFilterForAuthMessages \  
  --string-values device/${iot:ClientId}/auth
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Scoping metrics in security profiles using dimensions](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateDimension](#) Reference.

## update-domain-configuration

Il seguente esempio di codice mostra come utilizzare `update-domain-configuration`.

### AWS CLI

Per aggiornare la configurazione di un dominio

L'esempio seguente disattiva la configurazione di dominio specificata.

```
aws iot update-domain-configuration \  
  --domain-configuration-name "additionalDataDomain" \  
  --domain-configuration-status "DISABLED"
```

Output:

```
{  
  "domainConfigurationName": "additionalDataDomain",  
  "domainConfigurationArn": "arn:aws:iot:us-  
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"  
}
```

Per ulteriori informazioni, consulta [Configurable Endpoints](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateDomainConfiguration](#) Reference.

## update-dynamic-thing-group

Il seguente esempio di codice mostra come utilizzare `update-dynamic-thing-group`.

### AWS CLI

Per aggiornare un gruppo di oggetti dinamico

L'esempio seguente aggiorna il gruppo di cose dinamico specificato. Fornisce una descrizione e aggiorna la stringa di query per modificare i criteri di appartenenza al gruppo.

```
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"
```

Output:

```
{  
  "version": 2  
}
```

Per ulteriori informazioni, consulta [Dynamic Thing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [UpdateDynamicThingGroup](#) in AWS CLI Command Reference.

## update-event-configurations

Il seguente esempio di codice mostra come utilizzare `update-event-configurations`.

### AWS CLI

Per mostrare quali tipi di eventi sono pubblicati

L'esempio seguente aggiorna la configurazione per abilitare i messaggi quando il certificato CA viene aggiunto, aggiornato o eliminato.

```
aws iot update-event-configurations \  
  --event-configurations "{\"CA_CERTIFICATE\":{\"Enabled\":true}}"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Event Messages](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [UpdateEventConfigurations](#) in AWS CLI Command Reference.

## update-indexing-configuration

Il seguente esempio di codice mostra come utilizzare `update-indexing-configuration`.

### AWS CLI

Per abilitare l'indicizzazione degli oggetti

L'esempio seguente abilita l'indicizzazione degli oggetti per supportare la ricerca nei dati di registro, nei dati shadow e nello stato della connettività degli oggetti utilizzando l'indice `_Things`. AWS

```
aws iot update-indexing-configuration \  
  --thing-indexing-configuration thingIndexingMode=REGISTRY_AND_SHADOW,thingConnectivityIndexingMode=STATUS
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing Thing Indexing](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [UpdateIndexingConfiguration](#) in AWS CLI Command Reference.

## update-job

Il seguente esempio di codice mostra come utilizzare `update-job`.

### AWS CLI

Per ottenere lo stato dettagliato di un lavoro

L'`update-job` esempio seguente ottiene lo stato dettagliato del lavoro il cui ID è `example-job-01`.

```
aws iot describe-job \  
  --job-id "example-job-01"
```

Output:

```
{  
  "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": {}
}
}

```

Per ulteriori informazioni, consulta [Creating and Managing Jobs \(CLI\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [UpdateJob AWS CLI Command Reference](#).

## update-mitigation-action

Il seguente esempio di codice mostra come utilizzare `update-mitigation-action`.

### AWS CLI

Per aggiornare un'azione di mitigazione

L'esempio seguente aggiorna l'azione di mitigazione specificata denominata `AddThingsToQuarantineGroupAction`, modifica il nome del gruppo di cose e lo imposta `overrideDynamicGroups` su `false`. È possibile verificare le modifiche utilizzando il `describe-mitigation-action` comando.

```

aws iot update-mitigation-action \
  --cli-input-json "{ \"actionName\": \"AddThingsToQuarantineGroupAction\",
  \"actionParams\": { \"addThingsToThingGroupParams\": {\"thingGroupNames\":
  [\"QuarantineGroup2\"],\"overrideDynamicGroups\": false}}}"

```

Output:

```

{
  "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
  AddThingsToQuarantineGroupAction",
  "actionId": "2fd2726d-98e1-4abf-b10f-09465ccd6bfa"
}

```

Per ulteriori informazioni, consulta [UpdateMitigationAction \(Mitigation Action Commands\)](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [UpdateMitigationAction](#) in AWS CLI Command Reference.

## update-provisioning-template

Il seguente esempio di codice mostra come utilizzare `update-provisioning-template`.

### AWS CLI

Per aggiornare un modello di provisioning

L'esempio seguente modifica la descrizione e l'arn del ruolo per il modello di provisioning specificato e abilita il modello.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [AWS IoT Secure Tunneling](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateProvisioningTemplate](#) Reference.

## update-role-alias

Il seguente esempio di codice mostra come utilizzare `update-role-alias`.

### AWS CLI

Per aggiornare un alias di ruolo

L'esempio seguente aggiorna l'alias del `LightBulbRole` ruolo.

```
aws iot update-role-alias \  
  --role-alias LightBulbRole \  
  --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

Output:

```
{  
  "roleAlias": "LightBulbRole",
```

```
"roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"
}
```

Per ulteriori informazioni, [UpdateRoleAlias](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [UpdateRoleAlias](#) in AWS CLI Command Reference.

## update-scheduled-audit

Il seguente esempio di codice mostra come utilizzare `update-scheduled-audit`.

### AWS CLI

Per aggiornare una definizione di audit pianificato

L'`update-scheduled-audit` esempio seguente modifica i nomi dei controlli di destinazione per un audit pianificato di AWS IoT Device Defender.

```
aws iot update-scheduled-audit \
  --scheduled-audit-name WednesdayCertCheck \
  --target-check-
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK REVOKED_CA_CERTIFICATE
```

Output:

```
{
  "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
WednesdayCertCheck"
}
```

Per ulteriori informazioni, consulta [Audit Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [UpdateScheduledAudit](#) in AWS CLI Command Reference.

## update-security-profile

Il seguente esempio di codice mostra come utilizzare `update-security-profile`.

### AWS CLI

Per modificare un profilo di sicurezza

L'update-security-profile esempio seguente aggiorna sia la descrizione che i comportamenti per un profilo di sicurezza AWS IoT Device Defender.

```
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}}]"
```

Output:

```
{
  "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  
}
```

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [UpdateSecurityProfile](#) in AWS CLI Command Reference.

## update-stream

Il seguente esempio di codice mostra come utilizzare `update-stream`.

### AWS CLI

Per aggiornare uno stream

L'`update-stream` esempio seguente aggiorna uno stream esistente. La versione dello stream viene incrementata di uno.

```
aws iot update-stream \  
  --cli-input-json file://update-stream.json
```

Contenuto di `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"
}
```

Output:

```
{
  "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
}
```

Per ulteriori informazioni, [UpdateStream](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [UpdateStream](#) in AWS CLI Command Reference.

## update-thing-group

Il seguente esempio di codice mostra come utilizzare `update-thing-group`.

### AWS CLI

Per aggiornare la definizione di un gruppo di oggetti

L'`update-thing-group` esempio seguente aggiorna la definizione per il gruppo di oggetti specificato, modificando la descrizione e due attributi.

```
aws iot update-thing-group \
  --thing-group-name HalogenBulbs \
  --thing-group-properties "thingGroupDescription=\"Halogen bulb group\",
attributePayload={attributes={Manufacturer=AnyCompany,wattage=60}}"
```

Output:

```
{
  "version": 2
}
```

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [UpdateThingGroup](#) in AWS CLI Command Reference.

## update-thing-groups-for-thing

Il seguente esempio di codice mostra come utilizzare `update-thing-groups-for-thing`.

### AWS CLI

Per modificare i gruppi a cui appartiene un oggetto

L'esempio seguente rimuove l'oggetto denominato `MyLightBulb` dal gruppo denominato `DeadBulbs` e lo aggiunge al gruppo denominato `replaceableItems` contemporaneamente.

```
aws iot update-thing-groups-for-thing \  
  --thing-name MyLightBulb \  
  --thing-groups-to-add "replaceableItems" \  
  --thing-groups-to-remove "DeadBulbs"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Thing Groups](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [UpdateThingGroupsForThing](#) in AWS CLI Command Reference.

## update-thing

Il seguente esempio di codice mostra come utilizzare `update-thing`.

### AWS CLI

Associare un oggetto a un tipo di oggetto

L'esempio seguente associa un oggetto nel registro AWS IoT a un tipo di oggetto. Quando si crea l'associazione, si forniscono valori per gli attributi definiti dal tipo di oggetto.

```
aws iot update-thing \  
  --thing-name "MyOtherLightBulb" \  
  --thing-type-name "LightBulb" \  
  --attribute-payload '{"attributes": {"wattage": "75", "model": "123"}}'
```

Questo comando non produce output. Utilizzate il `describe-thing` comando per vedere il risultato.

Per ulteriori informazioni, consulta [Thing Types](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [UpdateThing](#) in AWS CLI Command Reference.

## update-topic-rule-destination

Il seguente esempio di codice mostra come utilizzare `update-topic-rule-destination`.

### AWS CLI

Esempio 1: per abilitare la destinazione di una regola tematica

L'`update-topic-rule-destination` esempio seguente abilita il traffico verso la destinazione di una regola tematica.

```
aws iot update-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --status ENABLED
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Enabling a topic rule destination](#) nella AWS IoT Developer Guide.

Esempio 2: disabilitare la destinazione di una regola tematica

L'`update-topic-rule-destination` esempio seguente disabilita il traffico verso la destinazione di una regola tematica.

```
aws iot update-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --status DISABLED
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Disabilitazione di una destinazione di regole tematiche](#) nella AWS IoT Developer Guide.

Esempio 3: Per inviare un nuovo messaggio di conferma

L'`update-topic-rule-destination` seguente invia un nuovo messaggio di conferma per la destinazione di una regola tematica.

```
aws iot update-topic-rule-destination \  
  --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --status IN_PROGRESS
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Invio di un nuovo messaggio di conferma](#) nella AWS IoT Developer Guide.

- Per API i dettagli, consulta [UpdateTopicRuleDestination AWS CLI Command Reference](#).

## `validate-security-profile-behaviors`

Il seguente esempio di codice mostra come utilizzare `validate-security-profile-behaviors`.

AWS CLI

Esempio 1: per convalidare i parametri di comportamento per un profilo di sicurezza

L'`validate-security-profile-behaviors` seguente convalida un set di comportamenti ben formato e corretto per un profilo di sicurezza IoT Device AWS Defender.

```
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}}]"
```

Output:

```
{  
  "valid": true,  
  "validationErrors": []  
}
```

Esempio 2: per convalidare i parametri di comportamento errati per un profilo di sicurezza

L'`validate-security-profile-behaviors` seguente convalida un set di comportamenti che contiene un errore per un profilo di sicurezza AWS IoT Device Defender.

```
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}}]"
```

Output:

```
{
  "valid": false,
  "validationErrors": [
    {
      "errorMessage": "Behavior Authorization is malformed.
consecutiveDatapointsToAlarm 100000 should be in range[1,10]"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Detect Commands](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [ValidateSecurityProfileBehaviors](#) in AWS CLI Command Reference.

## AWS IoT 1-Click Esempi di dispositivi che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS IoT 1-Click Devices.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **claim-devices-by-claim-code**

Il seguente esempio di codice mostra come utilizzare `claim-devices-by-claim-code`.

#### AWS CLI

Per richiedere uno o più dispositivi AWS IoT 1-Click utilizzando un codice promozionale

L'esempio seguente dichiara che il dispositivo AWS IoT 1-Click specificato utilizza un codice di attestazione (anziché un ID dispositivo).

```
aws iot1click-devices claim-devices-by-claim-code \  
  --claim-code C-123EXAMPLE
```

Output:

```
{  
  "Total": 9  
  "ClaimCode": "C-123EXAMPLE"  
}
```

Per ulteriori informazioni, consulta [Uso di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [ClaimDevicesByClaimCodeReference](#).

### **describe-device**

Il seguente esempio di codice mostra come utilizzare `describe-device`.

#### AWS CLI

Per descrivere un dispositivo

L'esempio seguente descrive il dispositivo specificato.

```
aws iot1click-devices describe-device \  
  --device-id G030PM0123456789
```

Output:

```
{  
  "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": {}  
  }  
}
```

Per ulteriori informazioni, consulta [Uso di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [DescribeDevice](#)Reference.

## finalize-device-claim

Il seguente esempio di codice mostra come utilizzare `finalize-device-claim`.

### AWS CLI

Per finalizzare una richiesta di reclamo per un dispositivo AWS IoT 1-Click utilizzando un ID dispositivo

L'esempio seguente finalizza una richiesta di reclamo per il dispositivo IoT AWS 1-Click specificato utilizzando un ID dispositivo (anziché un codice di reclamo).

```
aws iot1click-devices finalize-device-claim \  
  --device-id G030PM0123456789
```



```
--device-id G030PM0123456789
```

Output:

```
{
  "State": "CLAIMED"
}
```

Per ulteriori informazioni, consulta [Uso di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [FinalizeDeviceClaim](#)Reference.

## get-device-methods

Il seguente esempio di codice mostra come utilizzare `get-device-methods`.

AWS CLI

Per elencare i metodi disponibili per un dispositivo

L'`get-device-methods`esempio seguente elenca i metodi disponibili per un dispositivo.

```
aws iot1click-devices get-device-methods \  
  --device-id G030PM0123456789
```

Output:

```
{
  "DeviceMethods": [
    {
      "MethodName": "getDeviceHealthParameters"
    },
    {
      "MethodName": "setDeviceHealthMonitorCallback"
    },
    {
      "MethodName": "getDeviceHealthMonitorCallback"
    },
    {
      "MethodName": "setOnClickCallback"
    }
  ]
}
```

```
    },  
    {  
      "MethodName": "getOnClickCallback"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Uso di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [GetDeviceMethods](#)Reference.

## **initiate-device-claim**

Il seguente esempio di codice mostra come utilizzare `initiate-device-claim`.

### AWS CLI

Per avviare una richiesta di reclamo per un dispositivo AWS IoT 1-Click utilizzando un ID dispositivo

L'`initiate-device-claim`esempio seguente avvia una richiesta di reclamo per il dispositivo AWS IoT 1-Click specificato utilizzando un ID dispositivo (anziché un codice di reclamo).

```
aws iot1click-devices initiate-device-claim \  
  --device-id G030PM0123456789
```

Output:

```
{  
  "State": "CLAIM_INITIATED"  
}
```

Per ulteriori informazioni, consulta [Uso di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [InitiateDeviceClaim](#)Reference.

## **invoke-device-method**

Il seguente esempio di codice mostra come utilizzare `invoke-device-method`.

## AWS CLI

Per richiamare un metodo di dispositivo su un dispositivo

L'`invoke-device-method`esempio seguente richiama il metodo specificato su un dispositivo.

```
aws iot1click-devices invoke-device-method \  
--cli-input-json file://invoke-device-method.json
```

Contenuto di `invoke-device-method.json`.

```
{  
  "DeviceId": "G030PM0123456789",  
  "DeviceMethod": {  
    "DeviceType": "device",  
    "MethodName": "getDeviceHealthParameters"  
  }  
}
```

Output:

```
{  
  "DeviceMethodResponse": "{\"remainingLife\": 99.8}"  
}
```

Per ulteriori informazioni, consulta [Uso di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [InvokeDeviceMethod](#)Reference.

## list-device-events

Il seguente esempio di codice mostra come utilizzare `list-device-events`.

### AWS CLI

Per elencare gli eventi di un dispositivo per un intervallo di tempo specificato

L'`list-device-events`esempio seguente elenca gli eventi del dispositivo specificato per l'intervallo di tempo specificato.

```
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
```

Output:

```
{
  "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}"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Uso di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [ListDeviceEvents](#) Reference.

## list-devices

Il seguente esempio di codice mostra come utilizzare `list-devices`.

## AWS CLI

Per elencare i dispositivi di un tipo specificato

L'`list-devices` esempio seguente elenca i dispositivi di un tipo specificato.

```
aws iot1click-devices list-devices \  
  --device-type button
```

Questo comando non produce alcun output.

Output:

```
{  
  "Devices": [  
    {  
      "remainingLife": 99.9,  
      "attributes": {  
        "arn": "arn:aws:iot1click:us-west-2:123456789012:devices/  
G030PM0123456789",  
        "type": "button",  
        "deviceId": "G030PM0123456789",  
        "enabled": false  
      }  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Usa di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [ListDevices](#) Reference.

## **list-tags-for-resource**

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

## AWS CLI

Per elencare i tag di un dispositivo

L'`list-tags-for-resource` esempio seguente elenca i tag per il dispositivo specificato.

```
aws iot1click-devices list-tags-for-resource \  
  --resource-arn "arn:aws:iot1click:us-west-2:012345678901:devices/  
G030PM0123456789"
```

Output:

```
{  
  "Tags": {  
    "Driver Phone": "123-555-0199",  
    "Driver": "Jorge Souza"  
  }  
}
```

Per ulteriori informazioni, consulta [Uso di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [ListTagsForResource](#) Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere tag a una AWS risorsa del dispositivo

L'`tag-resource` esempio seguente aggiunge due tag alla risorsa specificata.

```
aws iot1click-devices tag-resource \  
  --cli-input-json file://devices-tag-resource.json
```

Contenuto di `devices-tag-resource.json`.

```
{  
  "ResourceArn": "arn:aws:iot1click:us-west-2:123456789012:devices/  
G030PM0123456789",  
  "Tags": {  
    "Driver": "Jorge Souza",  
    "Driver Phone": "123-555-0199"  
  }  
}
```

```
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Uso di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [TagResource](#)Reference.

## **unclaim-device**

Il seguente esempio di codice mostra come utilizzare `unclaim-device`.

### AWS CLI

Per rimuovere (annullare la registrazione) di un dispositivo dal tuo account AWS

L'`unclaim-device`esempio seguente annulla (annulla la registrazione) del dispositivo specificato dal tuo account. AWS

```
aws iot1click-devices unclaim-device \  
  --device-id G030PM0123456789
```

Output:

```
{  
  "State": "UNCLAIMED"  
}
```

Per ulteriori informazioni, consulta [Uso di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [UnclaimDevice](#)Reference.

## **untag-resource**

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una AWS risorsa del dispositivo

L'untag-resourceesempio seguente rimuove i tag con i nomi `Driver Phone` e `Driver` dalla risorsa del dispositivo specificata.

```
aws iot1click-devices untag-resource \  
  --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/  
AnytownDumpsters" \  
  --tag-keys "Driver Phone" "Driver"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Uso di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [UntagResource](#)Reference.

## update-device-state

Il seguente esempio di codice mostra come utilizzare `update-device-state`.

### AWS CLI

Per aggiornare lo stato ```abilitato``` per un dispositivo

Quanto segue `update-device-state` imposta lo stato del dispositivo specificato su `enabled`

```
aws iot1click-devices update-device-state \  
  --device-id G030PM0123456789 \  
  --enabled
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Uso di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [UpdateDeviceState](#)Reference.

## AWS IoT 1-Click Esempi di progetti che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS IoT 1-Click Projects.



Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto nei relativi scenari.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **associate-device-with-placement**

Il seguente esempio di codice mostra come utilizzare `associate-device-with-placement`.

AWS CLI

Per associare un dispositivo AWS IoT 1-Click a un posizionamento esistente

L'esempio seguente associa il dispositivo AWS IoT 1-Click specificato a un posizionamento esistente.

```
aws iot1click-projects associate-device-with-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217 \  
  --device-template-name empty-dumpster-request \  
  --device-id G030PM0123456789
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [AssociateDeviceWithPlacement](#) Reference.

### **create-placement**

Il seguente esempio di codice mostra come utilizzare `create-placement`.

## AWS CLI

Per creare un posizionamento AWS IoT 1-Click per un progetto

L'`create-placement`esempio seguente crea un posizionamento AWS IoT 1-Click per il progetto specificato.

```
aws iot1click-projects create-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217 \  
  --attributes '{"location": "123 Any Street Anytown, USA 10001", "phone":  
  "123-456-7890"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [CreatePlacement](#)Reference.

## `create-project`

Il seguente esempio di codice mostra come utilizzare `create-project`.

## AWS CLI

Per creare un progetto AWS IoT 1-Click per zero o più posizionamenti

L'`create-project`esempio seguente crea un progetto AWS IoT 1-Click per un posizionamento.

```
aws iot1click-projects create-project -- file: //create-project.json cli-input.json
```

Contenuto di `create-project.json`.

```
{  
  "projectName": "AnytownDumpsters",  
  "description": "All dumpsters in the Anytown region.",  
  "placementTemplate": {  
    "defaultAttributes": {  
      "City" : "Anytown"  
    },  
    "deviceTemplates": {  
      "empty-dumpster-request" : {
```

```
        "deviceType": "button"
      }
    }
  }
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [CreateProject](#)Reference.

## delete-placement

Il seguente esempio di codice mostra come utilizzare `delete-placement`.

### AWS CLI

Per eliminare un posizionamento da un progetto

L'`delete-placement`esempio seguente elimina il posizionamento specificato da un progetto.

```
aws iot1click-projects delete-placement \
  --project-name AnytownDumpsters \
  --placement-name customer217
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [DeletePlacement](#)Reference.

## delete-project

Il seguente esempio di codice mostra come utilizzare `delete-project`.

### AWS CLI

Per eliminare un progetto dal tuo AWS account

L'`delete-project`esempio seguente elimina il progetto specificato dal tuo AWS account.

```
aws iot1click-projects delete-project \  
  --project-name AnytownDumpsters
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [DeleteProjectReference](#).

## describe-placement

Il seguente esempio di codice mostra come utilizzare `describe-placement`.

### AWS CLI

Per descrivere il posizionamento di un progetto

L'esempio seguente descrive un posizionamento per il progetto specificato.

```
aws iot1click-projects describe-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217
```

Output:

```
{  
  "placement": {  
    "projectName": "AnytownDumpsters",  
    "placementName": "customer217",  
    "attributes": {  
      "phone": "123-555-0110",  
      "location": "123 Any Street Anytown, USA 10001"  
    },  
    "createdDate": 1563488454,  
    "updatedAt": 1563488454  
  }  
}
```

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [DescribePlacementReference](#).

## describe-project

Il seguente esempio di codice mostra come utilizzare `describe-project`.

### AWS CLI

Per descrivere un progetto AWS IoT 1-Click

L'`describe-project`esempio seguente descrive il progetto AWS IoT 1-Click specificato.

```
aws iot1click-projects describe-project \  
  --project-name AnytownDumpsters
```

Output:

```
{  
  "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": {}  
  }  
}
```

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [DescribeProjectReference](#).

## disassociate-device-from-placement

Il seguente esempio di codice mostra come utilizzare `disassociate-device-from-placement`.

### AWS CLI

Dissociare un dispositivo da un posizionamento

L'`disassociate-device-from-placement` esempio seguente dissocia il dispositivo specificato da un posizionamento.

```
aws iot1click-projects disassociate-device-from-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217 \  
  --device-template-name empty-dumpster-request
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [DisassociateDeviceFromPlacementReference](#).

## get-devices-in-placement

Il seguente esempio di codice mostra come utilizzare `get-devices-in-placement`.

### AWS CLI

Per elencare tutti i dispositivi in una posizione contenuta in un progetto

L'`get-devices-in-placement` esempio seguente elenca tutti i dispositivi in una posizione specificata contenuta nel progetto specificato.

```
aws iot1click-projects get-devices-in-placement \  
  --project-name AnytownDumpsters \  
  --placement-name customer217
```

Output:

```
{
  "devices": {
    "empty-dumpster-request": "G030PM0123456789"
  }
}
```

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [GetDevicesInPlacementReference](#).

## list-placements

Il seguente esempio di codice mostra come utilizzare `list-placements`.

### AWS CLI

Per elencare tutti i posizionamenti AWS IoT 1-Click per un progetto

L'`list-placements` esempio seguente elenca tutti i posizionamenti AWS IoT 1-Click per il progetto specificato.

```
aws iot1click-projects list-placements \
  --project-name AnytownDumpsters
```

Output:

```
{
  "placements": [
    {
      "projectName": "AnytownDumpsters",
      "placementName": "customer217",
      "createdDate": 1563488454,
      "updatedAt": 1563488454
    }
  ]
}
```

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [ListPlacementsReference](#).

## list-projects

Il seguente esempio di codice mostra come utilizzare `list-projects`.

### AWS CLI

Per elencare tutti i progetti AWS IoT 1-Click

L'`list-projects`esempio seguente elenca tutti i progetti AWS IoT 1-Click presenti nel tuo account.

```
aws iot1click-projects list-projects
```

Output:

```
{
  "projects": [
    {
      "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/
AnytownDumpsters",
      "projectName": "AnytownDumpsters",
      "createdDate": 1563483100,
      "updatedDate": 1563483100,
      "tags": {}
    }
  ]
}
```

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [ListProjectsReference](#).

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una risorsa del progetto

L'`list-tags-for-resource`esempio seguente elenca i tag per la risorsa di progetto specificata.



```
aws iot1click-projects list-tags-for-resource \  
  --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/  
AnytownDumpsters"
```

Output:

```
{  
  "tags": {  
    "Manager": "Li Juan",  
    "Account": "45215"  
  }  
}
```

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [ListTagsForResource](#) Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere tag a una risorsa del progetto

L'`tag-resource` esempio seguente aggiunge due tag alla risorsa di progetto specificata.

```
aws iot1click-projects tag-resource \  
  --cli-input-json file://devices-tag-resource.json
```

Contenuto di `devices-tag-resource.json`.

```
{  
  "resourceArn": "arn:aws:iot1click:us-west-2:123456789012:projects/  
AnytownDumpsters",  
  "tags": {  
    "Account": "45215",  
    "Manager": "Li Juan"  
  }  
}
```



```
--cli-input-json file://update-placement.json
```

Contenuto di `update-placement.json`.

```
{
  "projectName": "AnytownDumpsters",
  "placementName": "customer217",
  "attributes": {
    "phone": "123-456-7890",
    "location": "123 Any Street Anytown, USA 10001"
  }
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [UpdatePlacement](#)Reference.

## update-project

Il seguente esempio di codice mostra come utilizzare `update-project`.

AWS CLI

Per aggiornare le impostazioni di un progetto

L'`update-project`esempio seguente aggiorna la descrizione di un progetto.

```
aws iot1click-projects update-project \  
  --project-name AnytownDumpsters \  
  --description "All dumpsters (yard waste, recycling, garbage) in the Anytown  
region."
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Utilizzo di AWS IoT 1-Click con la Guida](#) per AWS CLI sviluppatori IoT AWS 1-Click.

- Per API i dettagli, consulta AWS CLI Command [UpdateProject](#)Reference.

## AWS IoT Analytics esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS IoT Analytics.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

### Azioni

#### batch-put-message

Il seguente esempio di codice mostra come utilizzare `batch-put-message`.

AWS CLI

Per inviare un messaggio a un canale

L'`batch-put-message` esempio seguente invia un messaggio al canale specificato.

```
aws iotanalytics batch-put-message \  
  --cli-binary-format raw-in-base64-out \  
  --cli-input-json file://batch-put-message.json
```

Contenuto di `batch-put-message.json`.

```
{  
  "channelName": "mychannel",  
  "messages": [  
    {  
      "messageId": "0001",  
      "payload": "eyJhdGVtcGVyYXR1cmUiOiAyMCMCB9"  
    }  
  ]  
}
```

```
]
}
```

Output:

```
{
  "batchPutMessageErrorEntries": []
}
```

Per ulteriori informazioni, [BatchPutMessage](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [BatchPutMessage AWS CLI](#) Command Reference.

## cancel-pipeline-reprocessing

Il seguente esempio di codice mostra come utilizzare `cancel-pipeline-reprocessing`.

AWS CLI

Per annullare la rielaborazione dei dati tramite una pipeline

L'esempio seguente annulla la rielaborazione dei dati tramite la pipeline specificata.

```
aws iotanalytics cancel-pipeline-reprocessing \
  --pipeline-name mypipeline \
  --reprocessing-id "6ad2764f-fb13-4de3-b101-4e74af03b043"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [CancelPipelineReprocessing](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [CancelPipelineReprocessing AWS CLI](#) Command Reference.

## create-channel

Il seguente esempio di codice mostra come utilizzare `create-channel`.

AWS CLI

Per creare un canale

L'create-channel esempio seguente crea un canale con la configurazione specificata. Un canale raccoglie dati da un MQTT argomento e archivia i messaggi non elaborati e non elaborati prima di pubblicarli in una pipeline.

```
aws iotanalytics create-channel \  
  --cli-input-json file://create-channel.json
```

Contenuto di create-channel.json.

```
{  
  "channelName": "mychannel",  
  "retentionPeriod": {  
    "unlimited": true  
  },  
  "tags": [  
    {  
      "key": "Environment",  
      "value": "Production"  
    }  
  ]  
}
```

Output:

```
{  
  "channelArn": "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel",  
  "channelName": "mychannel",  
  "retentionPeriod": {  
    "unlimited": true  
  }  
}
```

Per ulteriori informazioni, [CreateChannel](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [CreateChannel AWS CLI Command Reference](#).

## create-dataset-content

Il seguente esempio di codice mostra come utilizzare create-dataset-content.

## AWS CLI

Per creare il contenuto di un set di dati

L'`create-dataset-content` seguente crea il contenuto del set di dati specificato applicando a `queryAction` (una SQL query) o a `containerAction` (eseguendo un'applicazione containerizzata).

```
aws iotanalytics create-dataset-content \  
  --dataset-name mydataset
```

Output:

```
{  
  "versionId": "d494b416-9850-4670-b885-ca22f1e89d62"  
}
```

Per ulteriori informazioni, [CreateDatasetContent](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [CreateDatasetContent AWS CLI Command Reference](#).

## `create-dataset`

Il seguente esempio di codice mostra come utilizzare `create-dataset`.

## AWS CLI

Per creare un set di dati

L'`create-dataset` seguente crea un set di dati. Un set di dati memorizza i dati recuperati da un data store applicando a `queryAction` (una SQL query) o a `containerAction` (eseguendo un'applicazione containerizzata). Questa operazione crea lo scheletro di un set di dati. È possibile popolare il set di dati manualmente chiamando `CreateDatasetContent` o automaticamente in base a una specifica `trigger`

```
aws iotanalytics create-dataset \  
  --cli-input-json file://create-dataset.json
```

Contenuto di `create-dataset.json`.

```
{
```

```
"datasetName": "mydataset",
"actions": [
  {
    "actionName": "myDatasetAction",
    "queryAction": {
      "sqlQuery": "SELECT * FROM mydatastore"
    }
  }
],
"retentionPeriod": {
  "unlimited": true
},
"tags": [
  {
    "key": "Environment",
    "value": "Production"
  }
]
}
```

Output:

```
{
  "datasetName": "mydataset",
  "retentionPeriod": {
    "unlimited": true
  },
  "datasetArn": "arn:aws:iotanalytics:us-west-2:123456789012:dataset/mydataset"
}
```

Per ulteriori informazioni, [CreateDataset](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [CreateDataset AWS CLI](#) Command Reference.

## create-datastore

Il seguente esempio di codice mostra come utilizzare `create-datastore`.

### AWS CLI

Per creare un archivio dati

L'esempio seguente crea un data store, che è un archivio per i messaggi.



```
aws iotanalytics create-datastore \  
  --cli-input-json file://create-datastore.json
```

Contenuto di `create-datastore.json`.

```
{  
  "datastoreName": "mydatastore",  
  "retentionPeriod": {  
    "numberOfDays": 90  
  },  
  "tags": [  
    {  
      "key": "Environment",  
      "value": "Production"  
    }  
  ]  
}
```

Output:

```
{  
  "datastoreName": "mydatastore",  
  "datastoreArn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/  
mydatastore",  
  "retentionPeriod": {  
    "numberOfDays": 90,  
    "unlimited": false  
  }  
}
```

Per ulteriori informazioni, [CreateDatastore](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [CreateDatastore AWS CLI Command Reference](#).

## create-pipeline

Il seguente esempio di codice mostra come utilizzare `create-pipeline`.

### AWS CLI

Crea una pipeline IoT Analytics

L'`create-pipeline` seguente crea una pipeline. Una pipeline utilizza i messaggi provenienti da un canale e permette di elaborarli prima di archivarli in un datastore. È necessario specificare sia un canale che un'attività di archivio dati e, facoltativamente, fino a 23 attività aggiuntive nell'`pipelineActivities` array.

```
aws iotanalytics create-pipeline \  
--cli-input-json file://create-pipeline.json
```

Contenuto di `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"  
    }  
  ]  
}
```

## Output:

```
{
  "pipelineArn": "arn:aws:iotanalytics:us-west-2:123456789012:pipeline/
mypipeline",
  "pipelineName": "mypipeline"
}
```

Per ulteriori informazioni, [CreatePipeline](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [CreatePipeline AWS CLI Command Reference](#).

## delete-channel

Il seguente esempio di codice mostra come utilizzare `delete-channel`.

### AWS CLI

Eliminare un canale IoT Analytics

L'`delete-channel` esempio seguente elimina il canale specificato.

```
aws iotanalytics delete-channel \
  --channel-name mychannel
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [DeleteChannel](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [DeleteChannel AWS CLI Command Reference](#).

## delete-dataset-content

Il seguente esempio di codice mostra come utilizzare `delete-dataset-content`.

### AWS CLI

Per eliminare il contenuto del set di dati

L'`delete-dataset-content` esempio seguente elimina il contenuto del set di dati specificato.

```
aws iotanalytics delete-dataset-content \
  --dataset-name mydataset
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [DeleteDatasetContent](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [DeleteDatasetContent AWS CLI](#) Command Reference.

## delete-dataset

Il seguente esempio di codice mostra come utilizzare `delete-dataset`.

### AWS CLI

Per eliminare un set di dati

L'`delete-dataset` esempio seguente elimina il set di dati specificato. Non è necessario eliminare il contenuto del set di dati prima di eseguire questa operazione.

```
aws iotanalytics delete-dataset \  
  --dataset-name mydataset
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [DeleteDataset](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [DeleteDataset AWS CLI](#) Command Reference.

## delete-datastore

Il seguente esempio di codice mostra come utilizzare `delete-datastore`.

### AWS CLI

Per eliminare un archivio dati

L'`delete-datastore` esempio seguente elimina l'archivio dati specificato.

```
aws iotanalytics delete-datastore \  
  --datastore-name mydatastore
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [DeleteDatastore](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [DeleteDatastore AWS CLI Command Reference](#).

## delete-pipeline

Il seguente esempio di codice mostra come utilizzare `delete-pipeline`.

### AWS CLI

Per eliminare una pipeline

L'`delete-pipeline` esempio seguente elimina la tubazione specificata.

```
aws iotanalytics delete-pipeline \  
  --pipeline-name mypipeline
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [DeletePipeline](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [DeletePipeline AWS CLI Command Reference](#).

## describe-channel

Il seguente esempio di codice mostra come utilizzare `describe-channel`.

### AWS CLI

Per recuperare informazioni su un canale

L'`describe-channel` esempio seguente visualizza i dettagli, incluse le statistiche, per il canale specificato.

```
aws iotanalytics describe-channel \  
  --channel-name mychannel \  
  --include-statistics
```

Output:

```
{  
  "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"
    }
  }
}
```

Per ulteriori informazioni, [DescribeChannel](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [DescribeChannel AWS CLI](#) Command Reference.

## describe-dataset

Il seguente esempio di codice mostra come utilizzare `describe-dataset`.

### AWS CLI

Per recuperare informazioni su un set di dati

L'`describe-dataset` esempio seguente visualizza i dettagli per il set di dati specificato.

```
aws iotanalytics describe-dataset \
  --dataset-name mydataset
```

Output:

```
{
  "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"
}

```

Per ulteriori informazioni, [DescribeDataset](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [DescribeDataset AWS CLI Command Reference](#).

## describe-datastore

Il seguente esempio di codice mostra come utilizzare `describe-datastore`.

### AWS CLI

Per recuperare informazioni su un archivio dati

L'esempio seguente visualizza i dettagli, incluse le statistiche, per il data store specificato.

```

aws iotanalytics describe-datastore \
  --datastore-name mydatastore \
  --include-statistics

```

Output:

```

{
  "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
    }
  }
}
```

Per ulteriori informazioni, [DescribeDatastore](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [DescribeDatastore AWS CLI Command Reference](#).

## describe-logging-options

Il seguente esempio di codice mostra come utilizzare `describe-logging-options`.

### AWS CLI

Per recuperare le opzioni di registrazione correnti

L'esempio seguente mostra le opzioni di registrazione correnti di AWS IoT Analytics.

```
aws iotanalytics describe-logging-options
```

Questo comando non produce alcun output. Output:

```
{
  "loggingOptions": {
    "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",
    "enabled": true,
    "level": "ERROR"
  }
}
```

Per ulteriori informazioni, [DescribeLoggingOptions](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [DescribeLoggingOptions AWS CLI Command Reference](#).



## describe-pipeline

Il seguente esempio di codice mostra come utilizzare `describe-pipeline`.

### AWS CLI

Per recuperare informazioni su una pipeline

L'esempio seguente visualizza i dettagli della tubazione specificata.

```
aws iotanalytics describe-pipeline \  
  --pipeline-name mypipeline
```

Output:

```
{  
  "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"  
  }  
}
```

```
}  
}
```

Per ulteriori informazioni, [DescribePipeline](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [DescribePipeline AWS CLI](#) Command Reference.

## get-dataset-content

Il seguente esempio di codice mostra come utilizzare `get-dataset-content`.

### AWS CLI

Per recuperare il contenuto di un set di dati

L'`get-dataset-content` esempio seguente recupera il contenuto di un set di dati come predefinito. URIs

```
aws iotanalytics get-dataset-content --dataset-name mydataset
```

Output:

```
{  
  "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=..."  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta la guida [GetDatasetContent](#).

- Per API i dettagli, vedere [GetDatasetContent](#) in AWS CLI Command Reference.

## list-channels

Il seguente esempio di codice mostra come utilizzare `list-channels`.

### AWS CLI

Per recuperare un elenco di canali

L'`list-channels` esempio seguente visualizza informazioni di riepilogo per i canali disponibili.

```
aws iotanalytics list-channels
```

Output:

```
{
  "channelSummaries": [
    {
      "status": "ACTIVE",
      "channelName": "mychannel",
      "creationTime": 1557860351.001,
      "lastUpdateTime": 1557860351.001
    }
  ]
}
```

Per ulteriori informazioni, [ListChannels](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [ListChannels AWS CLI Command Reference](#).

## list-dataset-contents

Il seguente esempio di codice mostra come utilizzare `list-dataset-contents`.

### AWS CLI

Per elencare informazioni sul contenuto del set di dati

L'`list-dataset-contents` esempio seguente elenca le informazioni sui contenuti dei set di dati che sono stati creati.

```
aws iotanalytics list-dataset-contents \
  --dataset-name mydataset
```

**Output:**

```
{
  "datasetContentSummaries": [
    {
      "status": {
        "state": "SUCCEEDED"
      },
      "scheduleTime": 1557863215.995,
      "version": "b10ea2a9-66c1-4d99-8d1f-518113b738d0",
      "creationTime": 1557863215.995
    }
  ]
}
```

Per ulteriori informazioni, [ListDatasetContents](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [ListDatasetContents AWS CLI Command Reference](#).

**list-datasets**

Il seguente esempio di codice mostra come utilizzare `list-datasets`.

**AWS CLI**

Per recuperare informazioni sui set di dati

L'`list-datasets` esempio seguente elenca informazioni di riepilogo sui set di dati disponibili.

```
aws iotanalytics list-datasets
```

**Output:**

```
{
  "datasetSummaries": [
    {
      "status": "ACTIVE",
      "datasetName": "mydataset",
      "lastUpdateTime": 1557859240.658,
      "triggers": [],
      "creationTime": 1557859240.658,
      "actions": [
```

```
    {
      "actionName": "query_32",
      "actionType": "QUERY"
    }
  ]
}
```

Per ulteriori informazioni, [ListDatasets](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [ListDatasets AWS CLI](#) Command Reference.

## list-datastores

Il seguente esempio di codice mostra come utilizzare `list-datastores`.

### AWS CLI

Per recuperare un elenco di archivi dati

L'`list-datastores` esempio seguente visualizza informazioni di riepilogo sugli archivi dati disponibili.

```
aws iotanalytics list-datastores
```

Output:

```
{
  "datastoreSummaries": [
    {
      "status": "ACTIVE",
      "datastoreName": "mydatastore",
      "creationTime": 1557858971.02,
      "lastUpdateTime": 1557858971.02
    }
  ]
}
```

Per ulteriori informazioni, [ListDatastores](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [ListDatastores AWS CLI](#) Command Reference.

## list-pipelines

Il seguente esempio di codice mostra come utilizzare `list-pipelines`.

### AWS CLI

Per recuperare un elenco di pipeline

L'`list-pipelines` esempio seguente visualizza un elenco di tubazioni disponibili.

```
aws iotanalytics list-pipelines
```

Output:

```
{
  "pipelineSummaries": [
    {
      "pipelineName": "mypipeline",
      "creationTime": 1557859124.432,
      "lastUpdateTime": 1557859124.432,
      "reprocessingSummaries": []
    }
  ]
}
```

Per ulteriori informazioni, [ListPipelines](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [ListPipelines AWS CLI](#) Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una risorsa

L'`list-tags-for-resource` esempio seguente elenca i tag che avete allegato alla risorsa specificata.

```
aws iotanalytics list-tags-for-resource \
  --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel"
```

**Output:**

```
{
  "tags": [
    {
      "value": "bar",
      "key": "foo"
    }
  ]
}
```

Per ulteriori informazioni, [ListTagsForResource](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

**put-logging-options**

Il seguente esempio di codice mostra come utilizzare `put-logging-options`.

**AWS CLI**

Per impostare o aggiornare le opzioni di registrazione

L'`put-logging-options` esempio seguente imposta o aggiorna le opzioni di registrazione di AWS IoT Analytics. Se aggiorni il valore di un `loggingOptions` campo, può essere necessario fino a un minuto prima che la modifica abbia effetto. Inoltre, se modifichi la politica associata al ruolo specificato nel campo `roleArn` (ad esempio, per correggere una politica non valida), possono essere necessari fino a cinque minuti prima che la modifica abbia effetto.

```
aws iotanalytics put-logging-options \
  --cli-input-json file://put-logging-options.json
```

Contenuto di `put-logging-options.json`.

```
{
  "loggingOptions": {
    "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",
    "level": "ERROR",
    "enabled": true
  }
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [PutLoggingOptions](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [PutLoggingOptions AWS CLI Command Reference](#).

## run-pipeline-activity

Il seguente esempio di codice mostra come utilizzare `run-pipeline-activity`.

### AWS CLI

Per simulare un'attività di pipeline

L'`run-pipeline-activity` esempio seguente simula i risultati dell'esecuzione di un'attività di pipeline su un payload di messaggi.

```
aws iotanalytics run-pipeline-activity \  
  --pipeline-activity file://maths.json \  
  --payloads file://payloads.json
```

Contenuto di `maths.json`.

```
{  
  "math": {  
    "name": "MyMathActivity",  
    "math": "((temp - 32) * 5.0) / 9.0",  
    "attribute": "tempC"  
  }  
}
```

Contenuto di `payloads.json`.

```
[  
  "{\"humidity\": 52, \"temp\": 68 }",  
  "{\"humidity\": 52, \"temp\": 32 }"  
]
```

Output:

```
{
```



```
"logResult": "",
"payloads": [
  "eyJodW1pZG10eSI6NTIsInRlbXAiOjY4LCJ0ZW1wQyI6MjB9",
  "eyJodW1pZG10eSI6NTIsInRlbXAiOjMyLCJ0ZW1wQyI6MH0="
]
}
```

Per ulteriori informazioni, [RunPipelineActivity](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [RunPipelineActivity AWS CLI Command Reference](#).

## sample-channel-data

Il seguente esempio di codice mostra come utilizzare `sample-channel-data`.

### AWS CLI

Per recuperare messaggi di esempio da un canale

L'`sample-channel-data` esempio seguente recupera un campione di messaggi dal canale specificato importati durante il periodo di tempo specificato. È possibile recuperare fino a 10 messaggi.

```
aws iotanalytics sample-channel-data \
  --channel-name mychannel
```

Output:

```
{
  "payloads": [
    "eyJAidGVtcGVyYXR1cmUiOiAyMjB9",
    "eyJhZm9vIjogImJhcnVzIj0="
  ]
}
```

Per ulteriori informazioni, [SampleChannelData](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [SampleChannelData AWS CLI Command Reference](#).

## start-pipeline-reprocessing

Il seguente esempio di codice mostra come utilizzare `start-pipeline-reprocessing`.

## AWS CLI

Per avviare la rielaborazione della pipeline

L'`start-pipeline-reprocessing` seguente avvia la rielaborazione dei dati non elaborati dei messaggi attraverso la pipeline specificata.

```
aws iotanalytics start-pipeline-reprocessing \  
  --pipeline-name mypipeline
```

Output:

```
{  
  "reprocessingId": "6ad2764f-fb13-4de3-b101-4e74af03b043"  
}
```

Per ulteriori informazioni, [StartPipelineReprocessing](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [StartPipelineReprocessing AWS CLI Command Reference](#).

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere o modificare i tag di una risorsa

L'`tag-resource` seguente aggiunge o modifica i tag allegati alla risorsa specificata.

```
aws iotanalytics tag-resource \  
  --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \  
  --tags "[{"key": "Environment", "value": "Production"}]"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [TagResource](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [TagResource AWS CLI Command Reference](#).

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

## AWS CLI

Per rimuovere i tag da una risorsa

L'`untag-resource` esempio seguente rimuove i tag con i nomi di chiave specificati dalla risorsa specificata.

```
aws iotanalytics untag-resource \  
  --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \  
  --tag-keys ["\"Environment\""]
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta `UntagResource` < [https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API\\_UntagResource.html](https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UntagResource.html) > in AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [UntagResource AWS CLI Command Reference](#).

## update-channel

Il seguente esempio di codice mostra come utilizzare `update-channel`.

### AWS CLI

Per modificare un canale

L'`update-channel` esempio seguente modifica le impostazioni per il canale specificato.

```
aws iotanalytics update-channel \  
  --cli-input-json file://update-channel.json
```

Contenuto di `update-channel.json`.

```
{  
  "channelName": "mychannel",  
  "retentionPeriod": {  
    "numberOfDays": 92  
  }  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [UpdateChannel](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [UpdateChannel AWS CLI Command Reference](#).

## update-dataset

Il seguente esempio di codice mostra come utilizzare `update-dataset`.

### AWS CLI

Per aggiornare un set di dati

L'`update-dataset` esempio seguente modifica le impostazioni del set di dati specificato.

```
aws iotanalytics update-dataset \  
  --cli-input-json file://update-dataset.json
```

Contenuto di `update-dataset.json`.

```
{  
  "datasetName": "mydataset",  
  "actions": [  
    {  
      "actionName": "myDatasetUpdateAction",  
      "queryAction": {  
        "sqlQuery": "SELECT * FROM mydatastore"  
      }  
    }  
  ],  
  "retentionPeriod": {  
    "numberOfDays": 92  
  }  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta `UpdateDataset` < [https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API\\_UpdateDataset.html](https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UpdateDataset.html) > in AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [UpdateDataset AWS CLI Command Reference](#).

## update-datastore

Il seguente esempio di codice mostra come utilizzare `update-datastore`.

## AWS CLI

Per aggiornare un archivio dati

L'update-datastoreesempio seguente modifica le impostazioni del data store specificato.

```
aws iotanalytics update-datastore \  
  --cli-input-json file://update-datastore.json
```

Contenuto di update-datastore.json:

```
{  
  "datastoreName": "mydatastore",  
  "retentionPeriod": {  
    "numberOfDays": 93  
  }  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [UpdateDatastore](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [UpdateDatastore AWS CLI](#) Command Reference.

## update-pipeline

Il seguente esempio di codice mostra come utilizzareupdate-pipeline.

### AWS CLI

Per aggiornare una pipeline

L'update-pipelineesempio seguente modifica le impostazioni della tubazione specificata. È necessario specificare sia un canale che un'attività di archivio dati e, facoltativamente, fino a 23 attività aggiuntive, nell'array. pipelineActivities

```
aws iotanalytics update-pipeline \  
  --cli-input-json file://update-pipeline.json
```

Contenuto di 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"
    }
  }
]
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [UpdatePipeline](#) consulta AWS IoT Analytics API Reference.

- Per API i dettagli, consulta [UpdatePipeline AWS CLI Command Reference](#).

## Esempi di Device Advisor utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Device Advisor.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### create-suite-definition

Il seguente esempio di codice mostra come utilizzare `create-suite-definition`.

#### AWS CLI

Esempio 1: creare una suite di test IoT Device Advisor

L'esempio seguente crea una suite di test Device Advisor nell'AWS IoT con la configurazione di definizione della suite specificata.

```
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" }
```

Output:

```
{
  "suiteDefinitionId": "0jtsgio7yenu",
  "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/0jtsgio7yenu",
  "suiteDefinitionName": "TestSuiteName",
  "createdAt": "2022-12-02T11:38:13.263000-05:00"
}
```

Per ulteriori informazioni, consulta [Creare una definizione di suite di test](#) nella AWS IoT Core Developer Guide.

## Esempio 2: creare una suite di test di qualificazione più recente di IoT Device Advisor

L'`create-suite-definition` seguente crea una suite di test di qualificazione Device Advisor con la versione più recente dell' AWS IoT con la configurazione di definizione della suite specificata.

```
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"}
```

Output:

```
{
  "suiteDefinitionId": "txgsuolk2myj",
  "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/txgsuolk2myj",
  "suiteDefinitionName": "TestSuiteName",
  "createdAt": "2022-12-02T11:38:13.263000-05:00"
}
```

Per ulteriori informazioni, consulta [Creare una definizione di suite di test](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [CreateSuiteDefinition AWS CLI Command Reference](#).

## `delete-suite-definition`

Il seguente esempio di codice mostra come utilizzare `delete-suite-definition`.

### AWS CLI

Per eliminare la suite di test IoT Device Advisor

L'`delete-suite-definition` seguente elimina la suite di test Device Advisor con l'ID di definizione della suite specificato.

```
aws iotdeviceadvisor delete-suite-definition \
```



```
--suite-definition-id 0jtsgio7yenu
```

Questo comando non produce alcun output.

Per ulteriori informazioni, [DeleteSuiteDefinition](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [DeleteSuiteDefinition](#) in AWS CLI Command Reference.

## get-endpoint

Il seguente esempio di codice mostra come utilizzare `get-endpoint`.

### AWS CLI

Esempio 1: per ottenere informazioni su un endpoint a livello di account IoT Device Advisor

L'`get-endpoint` esempio seguente ottiene le informazioni su un endpoint di test a livello di account Device Advisor.

```
aws iotdeviceadvisor get-endpoint
```

Output:

```
{
  "endpoint": "t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

Esempio 2: per ottenere informazioni su un endpoint a livello di dispositivo IoT Device Advisor

L'`get-endpoint` esempio seguente ottiene le informazioni su un endpoint di test a livello di dispositivo Device Advisor con il `thing-arn` o `certificate-arn` specificato.

```
aws iotdeviceadvisor get-endpoint \
  --thing-arn arn:aws:iot:us-east-1:123456789012:thing/MyIoTThing
```

Output:

```
{
  "endpoint": "tdb7719be5t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

Per ulteriori informazioni, consulta [Get a test endpoint](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [GetEndpoint AWS CLI Command Reference](#).

## get-suite-definition

Il seguente esempio di codice mostra come utilizzare `get-suite-definition`.

### AWS CLI

Per ottenere informazioni su una suite di test IoT Device Advisor

L'`get-suite-definition` seguente ottiene le informazioni su una suite di test a device advisor con l'ID di definizione della suite specificato.

```
aws iotdeviceadvisor get-suite-definition \  
  --suite-definition-id qqcsmtyyjabl
```

Output:

```
{  
  "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\": \"uta5d9j1kvwc\",  
  \"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": {}  
}
```

Per ulteriori informazioni, consulta [Get a test suite definition](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [GetSuiteDefinition AWS CLI Command Reference](#).

## get-suite-run-report

Il seguente esempio di codice mostra come utilizzare `get-suite-run-report`.

### AWS CLI

Per ottenere informazioni sull'esecuzione di una suite di test qualificata per IoT Device Advisor

L'`get-suite-run-report` seguente ottiene il link per il download del rapporto relativo all'esecuzione di una suite di test qualificata per Device Advisor con successo con l'ID di definizione della suite e l'ID di esecuzione della suite specificati.

```
aws iotdeviceadvisor get-suite-run-report \  
  --suite-definition-id ztvb5aek4w4x \  
  --suite-run-id p6awv83nre6v
```

Output:

```
{  
  "qualificationReportDownloadUrl": "https://senate-apn-reports-us-east-1-  
prod.s3.amazonaws.com/report.downloadlink"  
}
```

Per ulteriori informazioni, consulta [Ottenere un rapporto di qualificazione per una suite di test di qualificazione eseguita con successo](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetSuiteRunReport Reference](#).

## get-suite-run

Il seguente esempio di codice mostra come utilizzare `get-suite-run`.

### AWS CLI

Per ottenere informazioni sullo stato di esecuzione di una suite di test IoT Device Advisor

L'`get-suite-run` seguente ottiene le informazioni sullo stato di esecuzione di una suite di test Device Advisor con l'ID di definizione della suite e l'ID di esecuzione della suite specificati.

```
aws iotdeviceadvisor get-suite-run \
  --suite-definition-id qqcsmtyyjabl \
  --suite-run-id nzlfyhaa18oa
```

## Output:

```
{
  "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": {}
}
```

Per ulteriori informazioni, consulta [Get a test suite run](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [GetSuiteRun AWS CLI Command Reference](#).

## list-suite-definitions

Il seguente esempio di codice mostra come utilizzare `list-suite-definitions`.

### AWS CLI

Esempio 1: per elencare le suite di test IoT Device Advisor che hai creato

L'`list-suite-definitions` esempio seguente elenca fino a 25 suite di test Device Advisor create in AWS IoT. Se disponi di più di 25 suite di test, nell'output verrà visualizzato `nextToken` "". Puoi usare questo "nextToken" per mostrare il resto delle suite di test che hai creato.

```
aws iotdeviceadvisor list-suite-definitions
```

Output:

```
{
  "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"
}

```

Esempio 2: per elencare le suite di test IoT Device Advisor create con le impostazioni specificate

L'`list-suite-definitions` seguente elenca le suite di test Device Advisor create in AWS IoT con il numero massimo di risultati specificato. Se disponi di più suite di test rispetto al numero massimo, nell'`nextToken` output verrà visualizzato "". Se hai "nextToken", puoi usare "nextToken" per mostrare le suite di test che hai creato e che non erano mai state mostrate prima.

```

aws iotdeviceadvisor list-suite-definitions \
  --max-result 1 \
  --next-token "nextTokenValue"

```

Output:

```

{
  "suiteDefinitionInformationList": [
    {
      "suiteDefinitionId": "ztlv5aew4w4x",
      "suiteDefinitionName": "TestSuite2",
      "defaultDevices": [],
      "intendedForQualification": true,
      "isLongDurationTest": false,
      "protocol": "MqttV3_1_1",
      "createdAt": "2022-11-17T14:15:56.830000-05:00"
    }
  ],
  "nextToken": "nextTokenValue"
}

```

Per ulteriori informazioni, [ListSuiteDefinitions](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [ListSuiteDefinitions](#) in AWS CLI Command Reference.

## list-suite-runs

Il seguente esempio di codice mostra come utilizzare `list-suite-runs`.

## AWS CLI

Esempio 1: per elencare tutte le informazioni sullo stato di esecuzione della suite di test IoT Device Advisor specificata

L'`list-suite-runs` seguente elenca tutte le informazioni sullo stato di esecuzione della suite di test Device Advisor con l'ID di definizione della suite specificato. Se sono state eseguite più di 25 suite di test, nell'output verrà visualizzato `nextToken` "". Puoi usare questo "nextToken" per mostrare il resto delle esecuzioni della suite di test.

```
aws iotdeviceadvisor list-suite-runs \  
--suite-definition-id ztvb5aew4w4x
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

Esempio 2: per elencare le informazioni sulla suite di test IoT Device Advisor specificata, esegue lo stato con le impostazioni specificate

L'`list-suite-runs` seguente elenca le informazioni sullo stato di esecuzione della suite di test Device Advisor con l'ID di definizione della suite specificato e il numero massimo di risultati specificato. Se il numero di esecuzioni della suite di test è superiore al numero massimo, nell'output verrà visualizzato `nextToken` "". Se hai "nextToken", puoi usare "nextToken" per mostrare le esecuzioni della suite di test che non erano state mostrate prima.

```
aws iotdeviceadvisor list-suite-runs \  
  --suite-definition-id qqcsmtyyjaml \  
  --max-result 1 \  
  --next-token "nextTokenValue"
```

Output:

```
{  
  "suiteRunsList": [  
    {  
      "suiteDefinitionId": "qqcsmtyyjaml",  
      "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"  
}
```

Per ulteriori informazioni, [ListSuiteRuns](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [ListSuiteRuns](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag allegati a una risorsa IoT Device Advisor

L'`list-tags-for-resource` esempio seguente elenca i tag allegati a una risorsa Device Advisor. La risorsa Device Advisor può essere `SuiteDefinition-Arn` o `SuiteRun-Arn`.

```
aws iotdeviceadvisor list-tags-for-resource \  
  --resource-arn arn:aws:iot:us-east-1:123456789012:suite-definition:my-suite-definition
```



```
--resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/  
ba0uyjpg38ny
```

Output:

```
{  
  "tags": {  
    "TestTagKey": "TestTagValue"  
  }  
}
```

Per ulteriori informazioni, vedere [ListTagsForResource AWS IoT API Reference](#) and [Resource types definiti da AWS IoT Core Device Advisor](#) nel Service Authorization Reference.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## start-suite-run

Il seguente esempio di codice mostra come utilizzare `start-suite-run`.

### AWS CLI

Per avviare l'esecuzione di una suite di test IoT Device Advisor

L'`start-suite-run` esempio seguente elenca i widget disponibili nel tuo AWS account.

```
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"}}'
```

Output:

```
{  
  "suiteRunId": "pwmucgw71t9s",  
  "suiteRunArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suiterun/  
qqcsmtyyjabl/pwmucgw71k9s",  
  "createdAt": "2022-12-02T15:43:05.581000-05:00"  
}
```

Per ulteriori informazioni, consulta [Avviare l'esecuzione di una suite di test](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [StartSuiteRun AWS CLI](#) Command Reference.

## stop-suite-run

Il seguente esempio di codice mostra come utilizzare `stop-suite-run`.

### AWS CLI

Per interrompere una suite di test IoT Device Advisor attualmente in esecuzione

L'`stop-suite-run` esempio seguente interrompe una suite di test Device Advisor attualmente in esecuzione con l'ID di definizione della suite e l'ID di esecuzione della suite specificati.

```
aws iotdeviceadvisor stop-suite-run \  
  --suite-definition-id qqcsmtyyjabl \  
  --suite-run-id nzlfyhaa18oa
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Stop a test suite run](#) nella AWS IoT Core Developer Guide.

- Per API i dettagli, consulta [StopSuiteRun AWS CLI](#) Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere e modificare i tag esistenti di una risorsa IoT Device Advisor

L'`tag-resource` esempio seguente aggiunge e modifica i tag esistenti di una risorsa Device Advisor con la risorsa `arn` e i tag specificati. La risorsa Device Advisor può essere `SuiteDefinition-Arn` o `SuiteRun-Arn`.

```
aws iotdeviceadvisor tag-resource \  
  --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/  
ba0uyjpg38ny \  
  --tags '{"TagKey": "TagValue"}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [TagResource AWS IoT API Reference](#) and [Resource types definiti da AWS IoT Core Device Advisor](#) nel Service Authorization Reference.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag esistenti da una risorsa IoT Device Advisor

L'`untag-resource` esempio seguente rimuove i tag esistenti da una risorsa Device Advisor con la risorsa `arn` e la chiave `tag` specificati. La risorsa Device Advisor può essere un `SuiteDefinitionArn` o un `SuiteRunArn`.

```
aws iotdeviceadvisor untag-resource \  
  --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/  
ba0uyjpg38ny \  
  --tag-keys "TagKey"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [UntagResource AWS IoT API Reference](#) and [Resource types definiti da AWS IoT Core Device Advisor](#) nel Service Authorization Reference.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-suite-definition

Il seguente esempio di codice mostra come utilizzare `update-suite-definition`.

### AWS CLI

Esempio 1: aggiornare una suite di test IoT Device Advisor

L'`update-suite-definition` esempio seguente aggiorna una suite di test Device Advisor nell' AWS IoT con l'ID di definizione della suite e la configurazione della definizione della suite specificati.

```
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" }
```

Output:

```
{
  "suiteDefinitionId": "3hsn88h4p2g5",
  "suiteDefinitionName": "TestSuiteName",
  "suiteDefinitionVersion": "v3",
  "createdAt": "2022-11-17T14:15:56.830000-05:00",
  "lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"
}
```

Esempio 2: aggiornare una suite di test di qualificazione IoT Device Advisor

L'update-suite-definitionesempio seguente aggiorna una suite di test di qualificazione Device Advisor nell' AWS IoT con l'ID di definizione della suite e la configurazione della definizione della suite specificati.

```
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" }
```

Output:

```
{
```

```
"suiteDefinitionId": "txgsuolk2myj",
"suiteDefinitionName": "TestSuiteName",
"suiteDefinitionVersion": "v3",
"createdAt": "2022-11-17T14:15:56.830000-05:00",
"lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"
}
```

Per ulteriori informazioni, [UpdateSuiteDefinition](#) consulta AWS IoT API Reference.

- Per API i dettagli, vedere [UpdateSuiteDefinition](#) in AWS CLI Command Reference.

## AWS IoT data esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS IoT data.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **delete-thing-shadow**

Il seguente esempio di codice mostra come utilizzare `delete-thing-shadow`.

#### AWS CLI

Per eliminare il documento shadow di un dispositivo

L'`delete-thing-shadow` esempio seguente elimina l'intero documento ombra per il dispositivo denominato `MyRPI`.

```
aws iot-data delete-thing-shadow \
  --thing-name MyRPI \
```

```
"output.txt"
```

Il comando non produce alcun output sullo schermo, ma `output.txt` contiene informazioni che confermano la versione e il timestamp del documento ombra eliminato.

```
{"version":2,"timestamp":1560270384}
```

Per ulteriori informazioni, consulta [Using Shadows](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [DeleteThingShadow](#) in AWS CLI Command Reference.

## get-thing-shadow

Il seguente esempio di codice mostra come utilizzare `get-thing-shadow`.

### AWS CLI

Per ottenere un oggetto, un documento ombra

L'`get-thing-shadow` seguente ottiene il documento shadow della cosa per l'oggetto IoT specificato.

```
aws iot-data get-thing-shadow \  
  --thing-name MyRPi \  
  output.txt
```

Il comando non produce alcun output sul display, ma quanto segue mostra il contenuto di `output.txt`:

```
{  
  "state":{  
    "reported":{  
      "moisture":"low"  
    }  
  },  
  "metadata":{  
    "reported":{  
      "moisture":{  
        "timestamp":1560269319  
      }  
    }  
  },  
}
```

```
"version":1,"timestamp":1560269405
}
```

Per ulteriori informazioni, consulta [Device Shadow Service Data Flow](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [GetThingShadow](#) in AWS CLI Command Reference.

## update-thing-shadow

Il seguente esempio di codice mostra come utilizzare `update-thing-shadow`.

### AWS CLI

Per aggiornare una cosa, shadow

L'`update-thing-shadow` esempio seguente modifica lo stato corrente del device shadow per l'oggetto specificato e lo salva nel file `output.txt`.

```
aws iot-data update-thing-shadow \
  --thing-name MyRPi \
  --payload '{"state":{"reported":{"moisture":"okay"}}}' \
  output.txt
```

Il comando non produce alcun output sul display, ma quanto segue mostra il contenuto di `output.txt`:

```
{
  "state": {
    "reported": {
      "moisture": "okay"
    }
  },
  "metadata": {
    "reported": {
      "moisture": {
        "timestamp": 1560270036
      }
    }
  },
  "version": 2,
  "timestamp": 1560270036
}
```

```
}
```

Per ulteriori informazioni, consulta [Device Shadow Service Data Flow](#) nella AWS IoT Developers Guide.

- Per API i dettagli, vedere [UpdateThingShadow](#) in AWS CLI Command Reference.

## AWS IoT Events esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS IoT Events.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **batch-put-message**

Il seguente esempio di codice mostra come utilizzare `batch-put-message`.

#### AWS CLI

Per inviare messaggi (input) a AWS IoT Events

L'`batch-put-message` esempio seguente invia una serie di messaggi al sistema AWS IoT Events. Ogni payload del messaggio viene trasformato nell'input specificato (`inputName`) e inserito in tutti i rilevatori che monitorano tale input. Se vengono inviati più messaggi, l'ordine in cui i messaggi vengono elaborati non è garantito. Per garantire l'ordine, è necessario inviare i messaggi uno alla volta e attendere una risposta corretta.

```
aws iotevents-data batch-put-message \  
  --cli-input-json file://highPressureMessage.json
```



## Contenuto di `highPressureMessage.json`.

```
{
  "messages": [
    {
      "messageId": "00001",
      "inputName": "PressureInput",
      "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\": 80, \"temperature\": 39} }"
    }
  ]
}
```

## Output:

```
{
  "BatchPutMessageErrorEntries": []
}
```

Per ulteriori informazioni, consulta [BatchPutMessage AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [BatchPutMessage](#) in AWS CLI Command Reference.

## **batch-update-detector**

Il seguente esempio di codice mostra come utilizzare `batch-update-detector`.

### AWS CLI

Per aggiornare un rilevatore (istanza)

L'`batch-update-detector` esempio seguente aggiorna lo stato, i valori delle variabili e le impostazioni del timer di uno o più rilevatori (istanze) di un modello di rilevatore specificato.

```
aws iotevents-data batch-update-detector \
  --cli-input-json file://budFulton-A32.json
```

## Contenuto di `budFulton-A32.json`.

```
{
  "detectors": [
```

```

    {
      "messageId": "00001",
      "detectorModelName": "motorDetectorModel",
      "keyValue": "Fulton-A32",
      "state": {
        "stateName": "Normal",
        "variables": [
          {
            "name": "pressureThresholdBreached",
            "value": "0"
          }
        ],
        "timers": [
        ]
      }
    }
  ]
}

```

Output:

```

{
  "batchUpdateDetectorErrorEntries": []
}

```

Per ulteriori informazioni, consulta [BatchUpdateDetector AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [BatchUpdateDetector](#) in AWS CLI Command Reference.

## create-detector-model

Il seguente esempio di codice mostra come utilizzare `create-detector-model`.

### AWS CLI

Per creare un modello di rilevatore

L'esempio seguente crea un modello di rivelatore con la sua configurazione specificata da un file di parametri.

```

aws iotevents create-detector-model \
  --cli-input-json file://motorDetectorModel.json

```

## Contenuto di 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": "$input.PressureInput.sensorData.pressure
                > 70",
                "actions": [
                  {
                    "setVariable": {
                      "variableName": "pressureThresholdBreached",
                      "value":
"$variable.pressureThresholdBreached + 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": "pressureThresholdBreach",
                  "value": "3"
                }
              }
            ]
          }
        ],
      },
      {
        "eventName": "Pressure Okay",
        "condition": "$input.PressureInput.sensorData.pressure
&lt;= 70",
        "actions": [
          {
            "setVariable": {
              "variableName": "pressureThresholdBreach",
              "value":
"$variable.pressureThresholdBreach - 1"

```

```

    }
  }
]
},
"transitionEvents": [
  {
    "eventName": "BackToNormal",
    "condition": "$input.PressureInput.sensorData.pressure
&lt;= 70 &amp;&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"
}

```

**Output:**

```

{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [CreateDetectorModel AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [CreateDetectorModel](#) in AWS CLI Command Reference.

## create-input

Il seguente esempio di codice mostra come utilizzare `create-input`.

### AWS CLI

Per creare un input

L'`create-input` esempio seguente crea un input.

```
aws iotevents create-input \
  --cli-input-json file://pressureInput.json
```

Contenuto di `pressureInput.json`.

```
{
  "inputName": "PressureInput",
  "inputDescription": "Pressure readings from a motor",
  "inputDefinition": {
    "attributes": [
      { "jsonPath": "sensorData.pressure" },
      { "jsonPath": "motorid" }
    ]
  }
}
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [CreateInput AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [CreateInput](#) in AWS CLI Command Reference.

## delete-detector-model

Il seguente esempio di codice mostra come utilizzare `delete-detector-model`.

### AWS CLI

Per eliminare un modello di rilevatore

L'esempio seguente elimina il modello di rilevatore specificato. Vengono eliminate anche tutte le istanze attive del modello di rilevatore.

```
aws iotevents delete-detector-model \
  --detector-model-name motorDetectorModel
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [DeleteDetectorModel AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [DeleteDetectorModel](#) in AWS CLI Command Reference.

## delete-input

Il seguente esempio di codice mostra come utilizzare `delete-input`.

### AWS CLI

Per eliminare un input

L'`delete-input` seguente elimina l'input specificato.

```
aws iotevents delete-input \  
  --input-name PressureInput
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [DeleteInput AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [DeleteInput](#) in AWS CLI Command Reference.

## `describe-detector-model`

Il seguente esempio di codice mostra come utilizzare `describe-detector-model`.

### AWS CLI

Per ottenere informazioni su un modello di rilevatore

L'`describe-detector-model` seguente visualizza i dettagli per il modello di rilevatore specificato. Poiché il `version` parametro non è specificato, vengono restituite informazioni sulla versione più recente.

```
aws iotevents describe-detector-model \  
  --detector-model-name motorDetectorModel
```

Output:

```
{  
  "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":
"pressureThresholdBreached",
                    "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.pressureThresholdBreached >
1"
                }
            ]
        },
        "onExit": {
            "events": [
                {
                    "eventName": "Normal Pressure Restored",
                    "actions": [
                        {
                            "sns": {
                                "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                            }
                        }
                    ],
                    "condition": "true"
                }
            ]
        }
    ],
    "initialStateName": "Normal"
}
}
}

```

Per ulteriori informazioni, consulta [DescribeDetectorModel](#) AWS IoT Events API Reference.

- Per API i dettagli, vedere [DescribeDetectorModel](#) in AWS CLI Command Reference.

## describe-detector

Il seguente esempio di codice mostra come utilizzare `describe-detector`.

### AWS CLI

Per ottenere informazioni su un rilevatore (istanza).

L'`describe-detector` seguente visualizza i dettagli per il rilevatore (istanza) specificato.

```
aws iotevents-data describe-detector \  
  --detector-model-name motorDetectorModel \  
  --key-value "Fulton-A32"
```

Output:

```
{  
  "detector": {  
    "lastUpdateTime": 1560797852.776,  
    "creationTime": 1560797852.775,  
    "state": {  
      "variables": [  
        {  
          "name": "pressureThresholdBreached",  
          "value": "3"  
        }  
      ],  
      "stateName": "Dangerous",  
      "timers": []  
    },  
    "keyValue": "Fulton-A32",  
    "detectorModelName": "motorDetectorModel",  
    "detectorModelVersion": "1"  
  }  
}
```

Per ulteriori informazioni, consulta [DescribeDetector AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [DescribeDetector](#) in AWS CLI Command Reference.

## describe-input

Il seguente esempio di codice mostra come utilizzare `describe-input`.

### AWS CLI

Per ottenere informazioni su un input

L'`describe-input`esempio seguente visualizza i dettagli per l'input specificato.

```
aws iotevents describe-input \  
  --input-name PressureInput
```

Output:

```
{  
  "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"  
        }  
      ]  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [DescribeInput AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [DescribeInput](#) in AWS CLI Command Reference.

## describe-logging-options

Il seguente esempio di codice mostra come utilizzare `describe-logging-options`.

### AWS CLI

Per ottenere informazioni sulle impostazioni di registrazione

L'`describe-logging-options` seguente recupera le impostazioni correnti delle opzioni di registrazione degli eventi AWS IoT.

```
aws iotevents describe-logging-options
```

Output:

```
{
  "loggingOptions": {
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
    "enabled": false,
    "level": "ERROR"
  }
}
```

Per ulteriori informazioni, consulta [DescribeLoggingOptions AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [DescribeLoggingOptions](#) in AWS CLI Command Reference.

## list-detector-model-versions

Il seguente esempio di codice mostra come utilizzare `list-detector-model-versions`.

### AWS CLI

Per ottenere informazioni sulle versioni di un modello di rilevatore

L'`list-detector-model-versions` seguente elenca tutte le versioni di un modello di rilevatore. Vengono restituiti solo i metadati associati a ciascuna versione del modello di rivelatore.

```
aws iotevents list-detector-model-versions \
  --detector-model-name motorDetectorModel
```

**Output:**

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [ListDetectorModelVersions AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [ListDetectorModelVersions](#) in AWS CLI Command Reference.

**list-detector-models**

Il seguente esempio di codice mostra come utilizzare `list-detector-models`.

**AWS CLI**

Per ottenere un elenco dei modelli di rilevatori

L'`list-detector-models` esempio seguente elenca i modelli di rilevatori che avete creato. Vengono restituiti solo i metadati associati a ciascun modello di rivelatore.

```
aws iotevents list-detector-models
```

**Output:**

```
{
  "detectorModelSummaries": [
    {
      "detectorModelName": "motorDetectorModel",
      "creationTime": 1552072424.212
      "detectorModelDescription": "Detect overpressure in a motor."
    }
  ]
}
```

```
    }
  ]
}
```

Per ulteriori informazioni, consulta [ListDetectorModels AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [ListDetectorModels](#) in AWS CLI Command Reference.

## list-detectors

Il seguente esempio di codice mostra come utilizzare `list-detectors`.

### AWS CLI

Per ottenere un elenco di rilevatori per un modello di rilevatore

L'`list-detectors` esempio seguente elenca i rilevatori (le istanze di un modello di rilevatore) presenti nel tuo account.

```
aws iotevents-data list-detectors \
  --detector-model-name motorDetectorModel
```

Output:

```
{
  "detectorSummaries": [
    {
      "lastUpdateTime": 1558129925.2,
      "creationTime": 1552073155.527,
      "state": {
        "stateName": "Normal"
      },
      "keyValue": "Fulton-A32",
      "detectorModelName": "motorDetectorModel",
      "detectorModelVersion": "1"
    }
  ]
}
```

Per ulteriori informazioni, consulta [ListDetectors AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [ListDetectors](#) in AWS CLI Command Reference.



## list-inputs

Il seguente esempio di codice mostra come utilizzare `list-inputs`.

### AWS CLI

Per elencare gli input

L'`list-inputs` esempio seguente elenca gli input che hai creato nel tuo account.

```
aws iotevents list-inputs
```

Questo comando non produce alcun output. Output:

```
{
  {
    "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"
  }
}
```

Per ulteriori informazioni, consulta [ListInputs AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [ListInputs](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag assegnati a una risorsa.

L'`list-tags-for-resource` esempio seguente elenca i nomi e i valori delle chiavi dei tag assegnati alla risorsa.

```
aws iotevents list-tags-for-resource \
```

```
--resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

Output:

```
{
  "tags": [
    {
      "value": "motor",
      "key": "deviceType"
    }
  ]
}
```

Per ulteriori informazioni, consulta [ListTagsForResource AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## put-logging-options

Il seguente esempio di codice mostra come utilizzare `put-logging-options`.

AWS CLI

Per impostare le opzioni di registrazione

L'operazione `put-logging-options` imposta o aggiorna le opzioni di registrazione degli eventi AWS IoT. Se si aggiorna il valore di un campo `loggingOptions`, il campo può richiedere fino a un minuto per la modifica di prendere effetto. Inoltre, se si cambia la politica associata al ruolo specificato nel campo `roleArn` (ad esempio, per correggere un criterio non valido), possono essere necessari fino a cinque minuti prima che la modifica abbia effetto.

```
aws iotevents put-logging-options \
  --cli-input-json file://logging-options.json
```

Contenuto di `logging-options.json`.

```
{
  "loggingOptions": {
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
```

```
    "level": "DEBUG",
    "enabled": true,
    "detectorDebugOptions": [
      {
        "detectorModelName": "motorDetectorModel",
        "keyValue": "Fulton-A32"
      }
    ]
  }
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [PutLoggingOptions AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [PutLoggingOptions](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere tag a una risorsa

L'`tag-resource` esempio seguente aggiunge o modifica (se la chiave esiste `deviceType` già) il tag associato alla risorsa specificata.

```
aws iotevents tag-resource \  
  --cli-input-json file://pressureInput.tag.json
```

Contenuto di `pressureInput.tag.json`.

```
{  
  "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",  
  "tags": [  
    {  
      "key": "deviceType",  
      "value": "motor"  
    }  
  ]  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [TagResource AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una risorsa

L'`untag-resource` esempio seguente rimuove il tag con il nome di chiave specificato dalla risorsa specificata.

```
aws iotevents untag-resource \  
  --resource-arn arn:aws:iotevents:us-west-2:123456789012:input/PressureInput \  
  --tagkeys deviceType
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [UntagResource AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-detector-model

Il seguente esempio di codice mostra come utilizzare `update-detector-model`.

### AWS CLI

Per aggiornare un modello di rilevatore

L'`update-detector-model` esempio seguente aggiorna il modello di rilevatore specificato. I rilevatori (istanze) generati dalla versione precedente vengono eliminati e quindi ricreati all'arrivo di nuovi input.

```
aws iotevents update-detector-model \  
  --cli-input-json file://motorDetectorModel.update.json
```

## Contenuto di motorDetectorModel.update.json.

```

{
  "detectorModelName": "motorDetectorModel",
  "detectorModelDefinition": {
    "states": [
      {
        "stateName": "Normal",
        "onEnter": {
          "events": [
            {
              "eventName": "init",
              "condition": "true",
              "actions": [
                {
                  "setVariable": {
                    "variableName": "pressureThresholdBreach",
                    "value": "0"
                  }
                }
              ]
            }
          ],
          "onInput": {
            "transitionEvents": [
              {
                "eventName": "Overpressurized",
                "condition": "$input.PressureInput.sensorData.pressure >
70",
                "actions": [
                  {
                    "setVariable": {
                      "variableName": "pressureThresholdBreach",
                      "value":
"$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.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"
}

```

### Output:

```

{
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [UpdateDetectorModel AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [UpdateDetectorModel](#) in AWS CLI Command Reference.

## update-input

Il seguente esempio di codice mostra come utilizzare `update-input`.

### AWS CLI

Per aggiornare un input

L'`update-input` seguente aggiorna l'input specificato con una nuova descrizione e definizione.

```
aws iotevents update-input \  
  --cli-input-json file://pressureInput.json
```

Contenuto di `pressureInput.json`.

```
{  
  "inputName": "PressureInput",  
  "inputDescription": "Pressure readings from a motor",  
  "inputDefinition": {  
    "attributes": [  
      { "jsonPath": "sensorData.pressure" },  
      { "jsonPath": "motorid" }  
    ]  
  }  
}
```

Output:



```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [UpdateInput AWS IoT Events API Reference](#).

- Per API i dettagli, vedere [UpdateInput](#) in AWS CLI Command Reference.

## AWS IoT Events-Data esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS IoT Events-Data.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **batch-put-message**

Il seguente esempio di codice mostra come utilizzare `batch-put-message`.

AWS CLI

Per inviare messaggi (input) a AWS IoT Events

L'batch-put-message esempio seguente invia una serie di messaggi al sistema AWS IoT Events. Ogni payload del messaggio viene trasformato nell'input specificato (`inputName`) e inserito in tutti i rilevatori che monitorano tale input. Se vengono inviati più messaggi, l'ordine in cui i messaggi vengono elaborati non è garantito. Per garantire l'ordine, è necessario inviare i messaggi uno alla volta e attendere una risposta corretta.

```
aws iotevents-data batch-put-message \  
  --cli-binary-format raw-in-base64-out \  
  --cli-input-json file://highPressureMessage.json
```

Contenuto di `highPressureMessage.json`.

```
{  
  "messages": [  
    {  
      "messageId": "00001",  
      "inputName": "PressureInput",  
      "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":  
80, \"temperature\": 39} }"  
    }  
  ]  
}
```

Output:

```
{  
  "BatchPutMessageErrorEntries": []  
}
```

Per ulteriori informazioni, consulta [BatchPutMessage](#) la AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [BatchPutMessage](#) in AWS CLI Command Reference.

## batch-update-detector

Il seguente esempio di codice mostra come utilizzare `batch-update-detector`.

### AWS CLI

Per aggiornare un rilevatore (istanza)

L'batch-update-detector esempio seguente aggiorna lo stato, i valori delle variabili e le impostazioni del timer di uno o più rilevatori (istanze) di un modello di rilevatore specificato.

```
aws iotevents-data batch-update-detector \  
  --cli-input-json file://budFulton-A32.json
```

Contenuto di budFulton-A32.json.

```
{  
  "detectors": [  
    {  
      "messageId": "00001",  
      "detectorModelName": "motorDetectorModel",  
      "keyValue": "Fulton-A32",  
      "state": {  
        "stateName": "Normal",  
        "variables": [  
          {  
            "name": "pressureThresholdBreached",  
            "value": "0"  
          }  
        ],  
        "timers": [  
        ]  
      }  
    }  
  ]  
}
```

Output:

```
{  
  "batchUpdateDetectorErrorEntries": []  
}
```

Per ulteriori informazioni, consulta [BatchUpdateDetector](#) nella AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [BatchUpdateDetector](#) in AWS CLI Command Reference.

## create-detector-model

Il seguente esempio di codice mostra come utilizzare `create-detector-model`.

## AWS CLI

Per creare un modello di rilevatore

L'create-detector-modelsempio seguente crea un modello di rilevatore.

```
aws iotevents create-detector-model \  
  --cli-input-json file://motorDetectorModel.json
```

Contenuto di 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": "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": "$input.PressureInput.sensorData.pressure
&lt;= 70",
            "actions": [
                {
                    "setVariable": {
                        "variableName": "pressureThresholdBreached",
                        "value":
"$variable.pressureThresholdBreached - 1"
                    }
                }
            ]
        },
        "transitionEvents": [
            {
                "eventName": "BackToNormal",
                "condition": "$input.PressureInput.sensorData.pressure
&lt;= 70 & & $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"
}
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [CreateDetectorModel](#) nella AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [CreateDetectorModel](#) in AWS CLI Command Reference.

## create-input

Il seguente esempio di codice mostra come utilizzare `create-input`.

### AWS CLI

Per creare un input

L'`create-input` esempio seguente crea un input.

```
aws iotevents create-input \
  --cli-input-json file://pressureInput.json
```

Contenuto di `pressureInput.json`.

```
{
  "inputName": "PressureInput",
  "inputDescription": "Pressure readings from a motor",
  "inputDefinition": {
```

```
    "attributes": [
      { "jsonPath": "sensorData.pressure" },
      { "jsonPath": "motorid" }
    ]
  }
}
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [CreateInput](#) nella AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [CreateInput](#) in AWS CLI Command Reference.

## delete-detector-model

Il seguente esempio di codice mostra come utilizzare `delete-detector-model`.

### AWS CLI

Per eliminare un modello di rilevatore

L'esempio seguente elimina un modello di rilevatore. Vengono eliminate anche tutte le istanze attive del modello di rilevatore.

```
aws iotevents delete-detector-model \
  --detector-model-name motorDetectorModel*
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [DeleteDetectorModel](#) nella AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [DeleteDetectorModel](#) in AWS CLI Command Reference.



## delete-input

Il seguente esempio di codice mostra come utilizzare `delete-input`.

### AWS CLI

Per eliminare un input

L'`delete-input` esempio seguente elimina un input.

```
aws iotevents delete-input \  
  --input-name PressureInput
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [DeleteInput](#) la AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [DeleteInput](#) in AWS CLI Command Reference.

## describe-detector-model

Il seguente esempio di codice mostra come utilizzare `describe-detector-model`.

### AWS CLI

Per ottenere informazioni su un modello di rilevatore

L'`describe-detector-model` esempio seguente descrive un modello di rilevatore. Se il `version` parametro non è specificato, il comando restituisce informazioni sulla versione più recente.

```
aws iotevents describe-detector-model \  
  --detector-model-name motorDetectorModel
```

Output:

```
{  
  "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":
"pressureThresholdBreached",
                    "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.pressureThresholdBreached >
1"
    }
  ],
  "onExit": {
    "events": [
      {
        "eventName": "Normal Pressure Restored",
        "actions": [
          {
            "sns": {
              "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
            }
          }
        ],
        "condition": "true"
      }
    ]
  }
}
],

```

```
        "initialStateName": "Normal"
      }
    }
  }
```

Per ulteriori informazioni, consulta [DescribeDetectorModel](#) nella AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [DescribeDetectorModel](#) in AWS CLI Command Reference.

## describe-detector

Il seguente esempio di codice mostra come utilizzare `describe-detector`.

### AWS CLI

Per ottenere informazioni su un rilevatore (istanza)

L'`describe-detector` seguente restituisce informazioni sul rilevatore (istanza) specificato.

```
aws iotevents-data describe-detector \
  --detector-model-name motorDetectorModel \
  --key-value "Fulton-A32"
```

Output:

```
{
  "detector": {
    "lastUpdateTime": 1560797852.776,
    "creationTime": 1560797852.775,
    "state": {
      "variables": [
        {
          "name": "pressureThresholdBreached",
          "value": "3"
        }
      ],
      "stateName": "Dangerous",
      "timers": []
    },
    "keyValue": "Fulton-A32",
    "detectorModelName": "motorDetectorModel",
```

```
    "detectorModelVersion": "1"
  }
}
```

Per ulteriori informazioni, consulta [DescribeDetector](#) nella AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [DescribeDetector](#) in AWS CLI Command Reference.

## describe-input

Il seguente esempio di codice mostra come utilizzare `describe-input`.

### AWS CLI

Per ottenere informazioni su un input

L'`describe-input` seguente recupera i dettagli di un input.

```
aws iotevents describe-input \
  --input-name PressureInput
```

Output:

```
{
  "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"
        }
      ]
    }
  }
}
```

```
    }  
  }  
}
```

Per ulteriori informazioni, consulta [DescribeInput](#)la AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [DescribeInput](#)in AWS CLI Command Reference.

## describe-logging-options

Il seguente esempio di codice mostra come utilizzare `describe-logging-options`.

### AWS CLI

Per ottenere informazioni sulle impostazioni di registrazione

L'`describe-logging-options`esempio seguente recupera le attuali opzioni di registrazione degli eventi AWS IoT.

```
aws iotevents describe-logging-options
```

Output:

```
{  
  "loggingOptions": {  
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
    "enabled": false,  
    "level": "ERROR"  
  }  
}
```

Per ulteriori informazioni, consulta [DescribeLoggingOptions](#)la AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [DescribeLoggingOptions](#)in AWS CLI Command Reference.

## list-detector-model-versions

Il seguente esempio di codice mostra come utilizzare `list-detector-model-versions`.

### AWS CLI

Per ottenere informazioni sulle versioni di un modello di rilevatore

L'`list-detector-model-versions` seguente elenca tutte le versioni di un modello di rilevatore. Vengono restituiti solo i metadati associati a ciascuna versione del modello di rivelatore.

```
aws iotevents list-detector-model-versions \  
--detector-model-name motorDetectorModel
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [ListDetectorModelVersions](#) la AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [ListDetectorModelVersions](#) in AWS CLI Command Reference.

## `list-detector-models`

Il seguente esempio di codice mostra come utilizzare `list-detector-models`.

AWS CLI

Per ottenere un elenco dei modelli di rilevatori

L'`list-detector-models` seguente elenca i modelli di rilevatori che avete creato. Vengono restituiti solo i metadati associati a ciascun modello di rivelatore.

```
aws iotevents list-detector-models
```

Output:



```
{
  "detectorModelSummaries": [
    {
      "detectorModelName": "motorDetectorModel",
      "creationTime": 1552072424.212
      "detectorModelDescription": "Detect overpressure in a motor."
    }
  ]
}
```

Per ulteriori informazioni, consulta [ListDetectorModels](#) la AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [ListDetectorModels](#) in AWS CLI Command Reference.

## list-detectors

Il seguente esempio di codice mostra come utilizzare `list-detectors`.

### AWS CLI

Per ottenere un elenco di rilevatori per un modello di rilevatore

L'`list-detectors` esempio seguente elenca i rilevatori (le istanze di un modello di rilevatore).

```
aws iotevents-data list-detectors \
  --detector-model-name motorDetectorModel
```

Output:

```
{
  "detectorSummaries": [
    {
      "lastUpdateTime": 1558129925.2,
      "creationTime": 1552073155.527,
      "state": {
        "stateName": "Normal"
      },
      "keyValue": "Fulton-A32",
      "detectorModelName": "motorDetectorModel",
      "detectorModelVersion": "1"
    }
  ]
}
```

```
}
```

Per ulteriori informazioni, consulta [ListDetectors](#) la AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [ListDetectors](#) in AWS CLI Command Reference.

## list-inputs

Il seguente esempio di codice mostra come utilizzare `list-inputs`.

### AWS CLI

Per elencare gli input

L'`list-inputs` esempio seguente elenca gli input che hai creato.

```
aws iotevents list-inputs
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [ListInputs](#) la AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [ListInputs](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag assegnati a una risorsa

L'`list-tags-for-resource` esempio seguente elenca i tag (metadati) assegnati alla risorsa.

```
aws iotevents list-tags-for-resource \  
  --resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

Output:

```
{  
  "tags": [  
    {  
      "value": "motor",  
      "key": "deviceType"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [ListTagsForResource](#) la AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## put-logging-options

Il seguente esempio di codice mostra come utilizzare `put-logging-options`.

AWS CLI

Per impostare le opzioni di registrazione

L'`list-tags-for-resource` esempio seguente imposta o aggiorna le opzioni di registrazione degli eventi AWS IoT. Se si aggiorna il valore di un `loggingOptions` campo, è necessario fino a un minuto prima che la modifica abbia effetto. Inoltre, se modifichi la politica associata al ruolo specificato nel `roleArn` campo (ad esempio, per correggere una politica non valida), occorrono fino a cinque minuti prima che la modifica abbia effetto.

```
aws iotevents put-logging-options \  
  --cli-input-json file://logging-options.json
```

Contenuto di `logging-options.json`.

```
{  
  "loggingOptions": {  
    "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",  
    "level": "DEBUG",
```

```
    "enabled": true,
    "detectorDebugOptions": [
      {
        "detectorModelName": "motorDetectorModel",
        "keyValue": "Fulton-A32"
      }
    ]
  }
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [PutLoggingOptions](#) nella AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [PutLoggingOptions](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere tag a una risorsa

L'`tag-resource` seguente aggiunge o modifica i tag della risorsa specificata. I tag sono metadati che possono essere utilizzati per gestire una risorsa.

```
aws iotevents tag-resource \
  --cli-input-json file://pressureInput.tag.json
```

Contenuto di `pressureInput.tag.json`.

```
{
  "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
  "tags": [
    {
      "key": "deviceType",
      "value": "motor"
    }
  ]
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [TagResource](#)la AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [TagResource](#)in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una risorsa

L'`untag-resource`esempio seguente rimuove i tag specificati dalla risorsa.

```
aws iotevents untag-resource \  
  --cli-input-json file://pressureInput.untag.json
```

Contenuto di `pressureInput.untag.json`.

```
{  
  "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",  
  "tagKeys": [  
    "deviceType"  
  ]  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [UntagResource](#)la AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [UntagResource](#)in AWS CLI Command Reference.

## update-detector-model

Il seguente esempio di codice mostra come utilizzare `update-detector-model`.

### AWS CLI

Per aggiornare un modello di rilevatore

L'`update-detector-model`esempio seguente aggiorna un modello di rilevatore. I rilevatori (istanze) generati dalla versione precedente vengono eliminati e quindi ricreati all'arrivo di nuovi input.

```
aws iotevents update-detector-model \  
  --cli-input-json file://motorDetectorModel.update.json
```

Contenuto motorDetectorModel di .update.json:

```
{  
  "detectorModelName": "motorDetectorModel",  
  "detectorModelDefinition": {  
    "states": [  
      {  
        "stateName": "Normal",  
        "onEnter": {  
          "events": [  
            {  
              "eventName": "init",  
              "condition": "true",  
              "actions": [  
                {  
                  "setVariable": {  
                    "variableName": "pressureThresholdBreach",  
                    "value": "0"  
                  }  
                }  
              ]  
            }  
          ]  
        },  
        "onInput": {  
          "transitionEvents": [  
            {  
              "eventName": "Overpressurized",  
              "condition": "$input.PressureInput.sensorData.pressure > 70",  
              "actions": [  
                {  
                  "setVariable": {  
                    "variableName": "pressureThresholdBreach",  
                    "value": "$variable.pressureThresholdBreach + 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"
},
"roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole"
}

```

### Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [UpdateDetectorModel](#) nella AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [UpdateDetectorModel](#) in AWS CLI Command Reference.

## update-input

Il seguente esempio di codice mostra come utilizzare `update-input`.

### AWS CLI

Per aggiornare un input

L'`update-input` seguente aggiorna un input.

```

aws iotevents update-input \
  --cli-input-json file://pressureInput.json

```

Contenuto di `pressureInput.json`.

```

{
  "inputName": "PressureInput",
  "inputDescription": "Pressure readings from a motor",
  "inputDefinition": {
    "attributes": [
      { "jsonPath": "sensorData.pressure" },
      { "jsonPath": "motorid" }
    ]
  }
}

```

Output:

```

{
  "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"
}
}
```

Per ulteriori informazioni, consulta [UpdateInput](#) la AWS IoT Events Developer Guide\*.

- Per API i dettagli, vedere [UpdateInput](#) in AWS CLI Command Reference.

## AWS IoT Greengrass esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS IoT Greengrass.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **associate-role-to-group**

Il seguente esempio di codice mostra come utilizzare `associate-role-to-group`.

#### AWS CLI

Associare un ruolo a un gruppo Greengrass

L'esempio seguente associa il IAM ruolo specificato a un gruppo Greengrass. Il ruolo di gruppo viene utilizzato dalle funzioni e dai connettori Lambda locali per accedere ai servizi AWS. Ad esempio, il ruolo del gruppo potrebbe concedere le autorizzazioni necessarie per l'integrazione dei CloudWatch log.

```
aws greengrass associate-role-to-group \  
  --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b \  
  --role-arn arn:aws:iam::123456789012:role/GG-Group-Role
```

Output:

```
{  
  "AssociatedAt": "2019-09-10T20:03:30Z"  
}
```

Per ulteriori informazioni, consulta [Configurare il ruolo del gruppo](#) nella AWS IoT Greengrass Developer Guide.

- Per API i dettagli, consulta [AssociateRoleToGroup AWS CLI Command Reference](#).

## associate-service-role-to-account

Il seguente esempio di codice mostra come utilizzare `associate-service-role-to-account`.

### AWS CLI

Per associare un ruolo di servizio al tuo AWS account

L'esempio seguente associa un ruolo IAM di servizio, specificato da `itsARN`, a AWS IoT Greengrass nel tuo AWS account. È necessario aver precedentemente creato il ruolo di servizio in IAM e associare ad esso un documento di policy che consenta a AWS IoT Greengrass di assumere questo ruolo.

```
aws greengrass associate-service-role-to-account \  
  --role-arn "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
```

Output:

```
{  
  "AssociatedAt": "2019-06-25T18:12:45Z"  
}
```

Per ulteriori informazioni, consulta [Greengrass Service Role](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, vedere [AssociateServiceRoleToAccount](#) in AWS CLI Command Reference.

## create-connector-definition-version

Il seguente esempio di codice mostra come utilizzare `create-connector-definition-version`.

### AWS CLI

Per creare una versione della definizione del connettore

L'esempio seguente crea una versione della definizione di connettore e la associa alla definizione di connettore specificata. Tutti i connettori di una versione definiscono i valori per i relativi parametri.

```
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-
  ntSlp6", "TwilioAuthTokenSecretArn-ResourceId": "TwilioAuthToken",
  "DefaultFromPhoneNumber": "4254492999"}]]"
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [CreateConnectorDefinitionVersion](#) in AWS CLI Command Reference.

## create-connector-definition

Il seguente esempio di codice mostra come utilizzare `create-connector-definition`.

### AWS CLI

Per creare una definizione di connettore

L'`create-connector-definition` seguente crea una definizione di connettore e una versione iniziale della definizione di connettore. La versione iniziale contiene un connettore. Tutti i connettori di una versione definiscono i valori per i relativi parametri.

```
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"}}}]'
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Getting Started with Greengrass Connectors \(CLI\)](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateConnectorDefinitionReference](#).

## create-core-definition-version

Il seguente esempio di codice mostra come utilizzare `create-core-definition-version`.

### AWS CLI

Per creare una versione di definizione principale

L'`create-core-definition-version` seguente crea una versione della definizione di base e la associa alla definizione di base specificata. La versione può contenere un solo core. Prima di poter creare un core, è necessario creare e fornire l'oggetto AWS IoT

corrispondente. Questo processo include i seguenti `iot` comandi, che restituiscono `ThingArn` e `CertificateArn` sono necessari per il `create-core-definition-version` comando.

Crea l'oggetto AWS IoT che corrisponde al dispositivo principale:

```
aws iot create-thing \  
  --thing-name "MyCoreDevice"
```

Output:

```
{  
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",  
  "thingName": "MyCoreDevice",  
  "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"  
}
```

Crea chiavi pubbliche e private e il certificato del dispositivo principale per l'oggetto. Questo esempio utilizza il `create-keys-and-certificate` comando e richiede le autorizzazioni di scrittura per la directory corrente. In alternativa, è possibile utilizzare il `create-certificate-from-csr` comando.

```
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"
```

Output:

```
{  
  "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-----
\nMIIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjBMXDcN0K4rMCxDR...fvY4+te\n-----END
RSA PRIVATE KEY-----\n"
  },
  "certificateId":
    "123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
}

```

Crea una policy AWS IoT che consenta iot e greengrass agisca. Per semplificare, la seguente politica consente azioni su tutte le risorse, ma la politica dovrebbe essere più restrittiva.

```

aws iot create-policy \
  --policy-name "Core_Devices" \
  --policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\n:\":\"Allow\", \"Action\":[\"iot:Publish\", \"iot:Subscribe\", \"iot:Connect
\n\", \"iot:Receive\"], \"Resource\":[\"*\"]}, {\"Effect\":[\"Allow\", \"Action\":[
\n\"iot:GetThingShadow\", \"iot:UpdateThingShadow\", \"iot>DeleteThingShadow\"],
\n\"Resource\":[\"*\"]}, {\"Effect\":[\"Allow\", \"Action\":[\"greengrass:*\"], \"Resource
\n\":[\"*\"]}]}"

```

Output:

```

{
  "policyName": "Core_Devices",
  "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/Core_Devices",
  "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\n:\":\"Allow\", \"Action\":[\"iot:Publish\", \"iot:Subscribe\", \"iot:Connect
\n\", \"iot:Receive\"], \"Resource\":[\"*\"]}, {\"Effect\":[\"Allow\", \"Action\":[
\n\"iot:GetThingShadow\", \"iot:UpdateThingShadow\", \"iot>DeleteThingShadow\"],
\n\"Resource\":[\"*\"]}, {\"Effect\":[\"Allow\", \"Action\":[\"greengrass:*\"], \"Resource
\n\":[\"*\"]}]}",
  "policyVersionId": "1"
}

```

Allega la politica al certificato:

```

aws iot attach-policy \
  --policy-name "Core_Devices" \
  --target "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"

```

Questo comando non produce alcun output.

Allega la cosa al certificato:

```
aws iot attach-thing-principal \  
  --thing-name "MyCoreDevice" \  
  --principal "arn:aws:iot:us-  
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Questo comando non produce alcun output.

Crea la versione di definizione principale:

```
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}]"
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Configurare AWS IoT Greengrass Core nella IoT AWS Greengrass Developer Guide](#).

- Per API i dettagli, consulta AWS CLI Command [CreateCoreDefinitionVersionReference](#).

## create-core-definition

Il seguente esempio di codice mostra come utilizzare `create-core-definition`.

### AWS CLI

Esempio 1: creare una definizione di base vuota



L'`create-core-definition` esempio seguente crea una definizione di base Greengrass vuota (nessuna versione iniziale). Prima che il core sia utilizzabile, è necessario utilizzare il `create-core-definition-version` comando per fornire gli altri parametri per il core.

```
aws greengrass create-core-definition \  
  --name cliGroup_Core
```

Output:

```
{  
  "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"  
}
```

Esempio 2: creare una definizione di base con una versione iniziale

L'`create-core-definition` esempio seguente crea una definizione di base che contiene una versione iniziale della definizione di base. La versione può contenere un solo core. Prima di poter creare un core, è necessario creare e fornire l'oggetto AWS IoT corrispondente. Questo processo include i seguenti `iot` comandi, che restituiscono `ThingArn` e `CertificateArn` sono necessari per il `create-core-definition` comando.

Crea l'oggetto AWS IoT che corrisponde al dispositivo principale:

```
aws iot create-thing \  
  --thing-name "MyCoreDevice"
```

Output:

```
{  
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",  
  "thingName": "MyCoreDevice",  
  "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"  
}
```

Crea chiavi pubbliche e private e il certificato del dispositivo principale per l'oggetto. Questo esempio utilizza il `create-keys-and-certificate` comando e richiede le autorizzazioni di

scrittura per la directory corrente. In alternativa, è possibile utilizzare il `create-certificate-from-csr` comando.

```
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"
```

Output:

```
{
  "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"
}
```

Crea una policy AWS IoT che consenta `iot` e `greengrass` agisca. Per semplificare, la seguente politica consente azioni su tutte le risorse, ma la politica dovrebbe essere più restrittiva.

```
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
\": [\"*\"]}]}"
```

Output:

```
{
  "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"
}
```

Allega la politica al certificato:

```
aws iot attach-policy \
  --policy-name "Core_Devices" \
  --target "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Questo comando non produce alcun output.

Allega la cosa al certificato:

```
aws iot attach-thing-principal \
  --thing-name "MyCoreDevice" \
  --principal "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

Questo comando non produce alcun output.

Crea la definizione principale:

```
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 } ] }"
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Configurare AWS IoT Greengrass Core nella IoT AWS Greengrass Developer Guide](#).

- Per API i dettagli, consulta AWS CLI Command [CreateCoreDefinitionReference](#).

## create-deployment

Il seguente esempio di codice mostra come utilizzare `create-deployment`.

### AWS CLI

Per creare una distribuzione per una versione di un gruppo Greengrass

L'`create-deployment` esempio seguente distribuisce la versione specificata di un gruppo Greengrass.

```
aws greengrass create-deployment \
  --deployment-type NewDeployment \
  --group-id "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca" \
  --group-version-id "dc40c1e9-e8c8-4d28-a84d-a9cad5f599c9"
```

Output:

```
{
  "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"
}
```

```
}
```

Per ulteriori informazioni, consulta [Getting Started with Connectors \(CLI\)](#) nella AWS IoT Greengrass Developer Guide.

- Per API i dettagli, vedere [CreateDeployment](#) in AWS CLI Command Reference.

## **create-device-definition-version**

Il seguente esempio di codice mostra come utilizzare `create-device-definition-version`.

### AWS CLI

Per creare una versione di definizione del dispositivo

L'`create-device-definition-version` esempio seguente crea una versione della definizione del dispositivo e la associa alla definizione del dispositivo specificata. La versione definisce due dispositivi. Prima di poter creare un dispositivo Greengrass, è necessario creare e fornire il dispositivo AWS IoT corrispondente. Questo processo include i seguenti `iot` comandi che è necessario eseguire per ottenere le informazioni richieste per il comando Greengrass:

Crea l'oggetto AWS IoT che corrisponde al dispositivo:

```
aws iot create-thing \  
  --thing-name "InteriorTherm"
```

Output:

```
{  
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",  
  "thingName": "InteriorTherm",  
  "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"  
}
```

Crea chiavi pubbliche e private e il certificato del dispositivo per l'oggetto. Questo esempio utilizza il `create-keys-and-certificate` comando e richiede le autorizzazioni di scrittura per la directory corrente. In alternativa, puoi usare il `create-certificate-from-csr` comando:

```
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"
```

**Output:**

```
{
  "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",
  "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCakGgAwIBATgIUCGq6EGqou6zFqWgIZRndgQEFW+gwDQYJKoZIhvc...KdGewQS\n-----END
CERTIFICATE-----\n",
  "keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBzrqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAAqKpRgnn6yq26U3y...wIDAQAB\n-----END
PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIIEowIABAKCAQEAAqKpRgnn6yq26U3yt5YFZquyukfRjBMXDcNOK4rMCxDR...fvY4+te\n-----END
RSA PRIVATE KEY-----\n"
  },
  "certificateId":
  "66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
}
```

Crea una policy AWS IoT che consenta `iot` e `greengrass` agisca. Per semplicità, la seguente politica consente azioni su tutte le risorse, ma la tua politica può essere più restrittiva:

```
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
\": [\"*\"]}]}"
```

**Output:**

```
{
  "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"
}

```

Allega la politica al certificato:

```

aws iot attach-policy \
  --policy-name "GG_Devices" \
  --target "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"

```

Allega l'oggetto al certificato

```

aws iot attach-thing-principal \
  --thing-name "InteriorTherm" \
  --principal "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"

```

Dopo aver creato e configurato l'oggetto IoT come mostrato sopra, usa i comandi ThingArn e CertificateArn dei primi due comandi nell'esempio seguente.

```

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}]"

```

Output:

```

{
  "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"  
}
```

- Per API i dettagli, consulta [CreateDeviceDefinitionVersion AWS CLI Command Reference](#).

## create-device-definition

Il seguente esempio di codice mostra come utilizzare `create-device-definition`.

### AWS CLI

Per creare una definizione di dispositivo

L'esempio seguente crea una definizione di dispositivo che contiene una versione iniziale della definizione del dispositivo. La versione iniziale definisce due dispositivi. Prima di poter creare un dispositivo Greengrass, è necessario creare e fornire il dispositivo AWS IoT corrispondente. Questo processo include i seguenti `iot` comandi che è necessario eseguire per ottenere le informazioni richieste per il comando Greengrass:

Crea l'oggetto AWS IoT che corrisponde al dispositivo:

```
aws iot create-thing \  
  --thing-name "InteriorTherm"
```

Output:

```
{  
  "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",  
  "thingName": "InteriorTherm",  
  "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"  
}
```

Crea chiavi pubbliche e private e il certificato del dispositivo per l'oggetto. Questo esempio utilizza il `create-keys-and-certificate` comando e richiede le autorizzazioni di scrittura per la directory corrente. In alternativa, puoi usare il `create-certificate-from-csr` comando:

```
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"
```

**Output:**

```
{
  "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92",
  "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-----
\nMIIIEowIABAKCAQEAqKpRgnn6yq26U3yt5YFZquyukfRjBMXDcNOK4rMCxDR...fvY4+te\n-----END
RSA PRIVATE KEY-----\n"
  },
  "certificateId":
  "66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
}
```

Crea una policy AWS IoT che consenta `iot` e `greengrass` agisca. Per semplicità, la seguente politica consente azioni su tutte le risorse, ma la tua politica può essere più restrittiva:

```
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
\": [\"*\"]}]}\""
```

**Output:**

```
{
  "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"
}

```

Allega la politica al certificato:

```

aws iot attach-policy \
  --policy-name "GG_Devices" \
  --target "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"

```

Allega l'oggetto al certificato

```

aws iot attach-thing-principal \
  --thing-name "InteriorTherm" \
  --principal "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"

```

Dopo aver creato e configurato l'oggetto IoT come mostrato sopra, usa i comandi ThingArn e CertificateArn dei primi due comandi nell'esempio seguente.

```

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\"},
{\"Id\": \"ExteriorTherm\", \"ThingArn\": \"arn:aws:iot:us-
west-2:123456789012:thing/ExteriorTherm\", \"CertificateArn\": \"arn:aws:iot:us-
west-2:123456789012:cert/6c52ce1b47bde88a637e9ccdd45fe4e4c2c0a75a6866f8f63d980ee22fa51e02\"},
{\"SyncShadow\": true}]}"}

```

Output:

```

{
  "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"
}

```

- Per API i dettagli, consulta [CreateDeviceDefinition AWS CLI Command Reference](#).

## create-function-definition-version

Il seguente esempio di codice mostra come utilizzare `create-function-definition-version`.

### AWS CLI

Per creare una versione della definizione della funzione

L'esempio seguente crea una nuova versione della definizione di funzione specificata. Questa versione specifica una singola funzione il cui ID è `Hello-World-function`, consente l'accesso al file system e specifica una dimensione massima della memoria e un periodo di timeout.

```

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}]}'

```

Output:

```

{
  "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"
}
```

- Per API i dettagli, vedere [CreateFunctionDefinitionVersion](#) in AWS CLI Command Reference.

## create-function-definition

Il seguente esempio di codice mostra come utilizzare `create-function-definition`.

### AWS CLI

Per creare una definizione di funzione Lambda

L'esempio seguente crea una definizione di funzione Lambda e una versione iniziale fornendo un elenco di funzioni Lambda (in questo caso, un elenco di una sola funzione denominata `TempMonitorFunction`) e le relative configurazioni. Prima di poter creare la definizione della funzione, è necessaria la funzione Lambda. ARN Per creare la funzione e il relativo alias, usa i comandi `create-function` e `publish-version` di Lambda. Il comando `create-function` di Lambda richiede il ruolo ARN di esecuzione, anche se AWS IoT Greengrass non utilizza quel ruolo perché le autorizzazioni sono specificate nel ruolo del gruppo Greengrass. Puoi usare il IAM `create-role` comando per creare un ruolo vuoto da utilizzare con Lambda `create-function` oppure puoi usare un ARN ruolo di esecuzione esistente.

```
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}}]}'
```

Output:

```
{
  "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"
}

```

Per ulteriori informazioni, consulta [Come configurare l'accesso alle risorse locali utilizzando l'interfaccia a riga di AWS comando](#) nella AWS IoT Greengrass Developer Guide.

- Per API i dettagli, vedere [CreateFunctionDefinition](#) in AWS CLI Command Reference.

## create-group-certificate-authority

Il seguente esempio di codice mostra come utilizzare `create-group-certificate-authority`.

### AWS CLI

Per creare un'autorità di certificazione (CA) per un gruppo

L'`create-group-certificate-authority` esempio seguente crea o ruota una CA per il gruppo specificato.

```

aws greengrass create-group-certificate-authority \
  --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"

```

Output:

```

{
  "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/8eaadd72-ce4b-4f15-892a-0cc4f3a343f1/certificateauthorities/
d31630d674c4437f6c5dbc0dca56312a902171ce2d086c38e509c8EXAMPLEecc5"
}

```

Per ulteriori informazioni, consulta [AWS IoT Greengrass Security nella IoT AWS Greengrass Developer Guide](#).

- Per API i dettagli, consulta AWS CLI Command [CreateGroupCertificateAuthority](#) Reference.

## create-group-version

Il seguente esempio di codice mostra come utilizzare `create-group-version`.

## AWS CLI

Per creare una versione di un gruppo Greengrass

L'`create-group-version` seguente crea una versione di gruppo e la associa al gruppo specificato. La versione fa riferimento alle versioni core, resource, connector, function e subscription che contengono le entità da includere in questa versione di gruppo. È necessario creare queste entità prima di poter creare la versione di gruppo.

Per creare una definizione di risorsa con una versione iniziale, utilizzare il `create-resource-definition` comando. Per creare una definizione di connettore con una versione iniziale, utilizzare il `create-connector-definition` comando. Per creare una definizione di funzione con una versione iniziale, utilizzare il `create-function-definition` comando. Per creare una definizione di sottoscrizione con una versione iniziale, utilizzare il `create-subscription-definition` comando. Per recuperare la versione più recente della definizione ARN di base, utilizzare il `get-group-version` comando e specificare l'ID dell'ultima versione di gruppo.

```
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"
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, vedere [Panoramica del modello a AWS oggetti del gruppo IoT Greengrass](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, vedere [CreateGroupVersion](#) in AWS CLI Command Reference.

## create-group

Il seguente esempio di codice mostra come utilizzare `create-group`.

### AWS CLI

Per creare un gruppo Greengrass

L'`create-group` esempio seguente crea un gruppo denominato `cli-created-group`

```
aws greengrass create-group \
  --name cli-created-group
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, vedere [Panoramica del modello a AWS oggetti del gruppo IoT Greengrass](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, vedere [CreateGroup](#) in AWS CLI Command Reference.

## create-logger-definition-version

Il seguente esempio di codice mostra come utilizzare `create-logger-definition-version`.

## AWS CLI

Per creare una versione della definizione del logger

L'`create-logger-definition-version` seguente crea una versione della definizione del logger e la associa a una definizione del logger. La versione definisce quattro configurazioni di registrazione: 1) registri dei componenti di sistema sul file system del dispositivo principale, 2) registri delle funzioni Lambda definiti dall'utente sul file system del dispositivo principale, 3) registri dei componenti di sistema in Amazon Logs e 4) registri delle funzioni Lambda definiti dall'utente in Amazon Logs CloudWatch. CloudWatch Nota: per l'integrazione con Logs, il ruolo del gruppo deve concedere le autorizzazioni appropriate. CloudWatch

```
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\\"}]"
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Monitoring with AWS IoT Greengrass Logs](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, vedere [CreateLoggerDefinitionVersion](#) in AWS CLI Command Reference.

### `create-logger-definition`

Il seguente esempio di codice mostra come utilizzare `create-logger-definition`.



## AWS CLI

Per creare una definizione di logger

L'`create-logger-definition` seguente crea una definizione di logger che contiene una versione iniziale della definizione del logger. La versione iniziale definisce tre configurazioni di registrazione: 1) registri dei componenti di sistema sul file system del dispositivo principale, 2) registri delle funzioni Lambda definiti dall'utente sul file system del dispositivo principale e 3) registri delle funzioni Lambda definiti dall'utente in Amazon Logs. CloudWatch Nota: per l'integrazione dei CloudWatch log, il ruolo del gruppo deve concedere le autorizzazioni appropriate.

```
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\"}]}"
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Monitoring with AWS IoT Greengrass Logs](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, vedere [CreateLoggerDefinition](#) in AWS CLI Command Reference.

## create-resource-definition-version

Il seguente esempio di codice mostra come utilizzare `create-resource-definition-version`.

### AWS CLI

Per creare una versione di una definizione di risorsa

L'esempio seguente crea una nuova versione di `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-nt5lp6\\\"}}}]"
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [CreateResourceDefinitionVersion](#) in AWS CLI Command Reference.

## create-resource-definition

Il seguente esempio di codice mostra come utilizzare `create-resource-definition`.

### AWS CLI

Per creare una definizione di risorsa

L'esempio seguente crea una definizione di risorsa che contiene un elenco di risorse da utilizzare in un gruppo Greengrass. In questo esempio, viene inclusa una versione iniziale della definizione delle risorse fornendo un elenco di risorse. L'elenco

include una risorsa per un token di autorizzazione Twilio e una ARN per un segreto archiviato in AWS Secrets Manager. È necessario creare il segreto prima di poter creare la definizione della risorsa.

```
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-ntS1p6\"}}}]}"
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Come configurare l'accesso alle risorse locali utilizzando l'interfaccia a riga di AWS comando](#) nella AWS IoT Greengrass Developer Guide.

- Per API i dettagli, vedere [CreateResourceDefinition](#) in AWS CLI Command Reference.

## create-software-update-job

Il seguente esempio di codice mostra come utilizzare `create-software-update-job`.

### AWS CLI

Per creare un processo di aggiornamento software per un core

L'esempio seguente crea un processo di aggiornamento over-the-air (OTA) per aggiornare il software AWS IoT Greengrass Core sul core il cui nome è `MyFirstGroup_Core`. Questo comando richiede un IAM ruolo che consenta l'accesso ai

pacchetti di aggiornamento software in Amazon S3 e lo includa `iot.amazonaws.com` come entità affidabile.

```
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
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [OTAAggiornamenti del software AWS IoT Greengrass Core](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, vedere [CreateSoftwareUpdateJob](#) in AWS CLI Command Reference.

## create-subscription-definition-version

Il seguente esempio di codice mostra come utilizzare `create-subscription-definition-version`.

### AWS CLI

Per creare una nuova versione di una definizione di abbonamento

L'esempio seguente crea una nuova versione di una definizione di abbonamento che contiene tre sottoscrizioni: una notifica di attivazione, un input di temperatura e uno stato di uscita.

```
aws greengrass create-subscription-definition-version \  
  --subscription-definition-id "9d611d57-5d5d-44bd-a3b4-fecbddd69112" \  
  --update-agent-log-level WARN
```

```
--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"}]"
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [CreateSubscriptionDefinitionVersion](#) in AWS CLI Command Reference.

## create-subscription-definition

Il seguente esempio di codice mostra come utilizzare `create-subscription-definition`.

### AWS CLI

Per creare una definizione di sottoscrizione

L'esempio seguente crea una definizione di sottoscrizione e ne specifica la versione iniziale. La versione iniziale contiene tre abbonamenti: uno per l'MQTT a cui è abbonato il connettore, uno per consentire a una funzione di ricevere letture della temperatura dall'AWS IoT e uno per consentire all'AWS IoT di ricevere informazioni sullo stato dal connettore. L'esempio fornisce l'ARN alias della funzione Lambda creato in precedenza utilizzando il comando di Lambda `create-alias`

```
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\"}}}]\"
```

Output:

```
{
  \"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\"
}
```

Per ulteriori informazioni, consulta [Getting Started with Connectors \(CLI\)](#) nella AWS IoT Greengrass Developer Guide.

- Per API i dettagli, vedere [CreateSubscriptionDefinition](#) in AWS CLI Command Reference.

## delete-connector-definition

Il seguente esempio di codice mostra come utilizzare `delete-connector-definition`.

### AWS CLI

Per eliminare una definizione di connettore

L'`delete-connector-definition`esempio seguente elimina la definizione del connettore Greengrass specificata. Se si elimina una definizione di connettore utilizzata da un gruppo, tale gruppo non può essere distribuito correttamente.

```
aws greengrass delete-connector-definition \
  --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [DeleteConnectorDefinition AWS CLI Command Reference](#).

## **delete-core-definition**

Il seguente esempio di codice mostra come utilizzare `delete-core-definition`.

### AWS CLI

Per eliminare una definizione di base

L'`delete-core-definition` seguente elimina la definizione principale Greengrass specificata, incluse tutte le versioni. Se elimini un core associato a un gruppo Greengrass, quel gruppo non può essere distribuito correttamente.

```
aws greengrass delete-core-definition \  
  --core-definition-id "ff36cc5f-9f98-4994-b468-9d9b6dc52abd"
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta AWS CLI Command [DeleteCoreDefinition Reference](#).

## **delete-device-definition**

Il seguente esempio di codice mostra come utilizzare `delete-device-definition`.

### AWS CLI

Per eliminare la definizione di un dispositivo

L'`delete-device-definition` seguente elimina la definizione del dispositivo specificata, incluse tutte le relative versioni. Se si elimina una versione di definizione del dispositivo utilizzata da una versione di gruppo, la versione di gruppo non può essere distribuita correttamente.

```
aws greengrass delete-device-definition \  
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteDeviceDefinition](#) in AWS CLI Command Reference.

## delete-function-definition

Il seguente esempio di codice mostra come utilizzare `delete-function-definition`.

### AWS CLI

Per eliminare la definizione di una funzione

L'esempio seguente elimina la definizione della funzione Greengrass specificata. Se si elimina una definizione di funzione utilizzata da un gruppo, tale gruppo non può essere distribuito correttamente.

```
aws greengrass delete-function-definition \  
  --function-definition-id "fd4b906a-dff3-4c1b-96eb-52ebfcfac06a"
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [DeleteFunctionDefinition AWS CLI Command Reference](#).

## delete-group

Il seguente esempio di codice mostra come utilizzare `delete-group`.

### AWS CLI

Per eliminare un gruppo

L'esempio seguente elimina il gruppo Greengrass specificato.

```
aws greengrass delete-group \  
  --group-id "4e22bd92-898c-436b-ade5-434d883ff749"
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteGroup](#) in AWS CLI Command Reference.

## delete-logger-definition

Il seguente esempio di codice mostra come utilizzare `delete-logger-definition`.

### AWS CLI

Per eliminare una definizione di logger



L'`delete-logger-definition` seguente elimina la definizione del logger specificata, incluse tutte le versioni della definizione del logger. Se si elimina una versione di definizione del logger utilizzata da una versione di gruppo, la versione di gruppo non può essere distribuita correttamente.

```
aws greengrass delete-logger-definition \  
  --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Monitoring with AWS IoT Greengrass Logs](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, vedere [DeleteLoggerDefinition](#) in AWS CLI Command Reference.

## **delete-resource-definition**

Il seguente esempio di codice mostra come utilizzare `delete-resource-definition`.

### AWS CLI

Per eliminare una definizione di risorsa

L'`delete-resource-definition` seguente elimina la definizione di risorsa specificata, incluse tutte le versioni della risorsa. Se si elimina una definizione di risorsa utilizzata da un gruppo, tale gruppo non può essere distribuito correttamente.

```
aws greengrass delete-resource-definition \  
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [DeleteResourceDefinition AWS CLI](#) Command Reference.

## **delete-subscription-definition**

Il seguente esempio di codice mostra come utilizzare `delete-subscription-definition`.

### AWS CLI

Per eliminare una definizione di sottoscrizione

L'`delete-subscription-definition` seguente elimina la definizione di abbonamento Greengrass specificata. Se si elimina un abbonamento utilizzato da un gruppo, tale gruppo non può essere distribuito correttamente.

```
aws greengrass delete-subscription-definition \  
  --subscription-definition-id "cd6f1c37-d9a4-4e90-be94-01a7404f5967"
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [DeleteSubscriptionDefinition AWS CLI Command Reference](#).

## **disassociate-role-from-group**

Il seguente esempio di codice mostra come utilizzare `disassociate-role-from-group`.

### AWS CLI

Dissociare il ruolo da un gruppo Greengrass

L'`disassociate-role-from-group` seguente dissocia il IAM ruolo dal gruppo Greengrass specificato.

```
aws greengrass disassociate-role-from-group \  
  --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

Output:

```
{  
  "DisassociatedAt": "2019-09-10T20:05:49Z"  
}
```

Per ulteriori informazioni, consulta [Configurare il ruolo del gruppo](#) nella AWS IoT Greengrass Developer Guide.

- Per API i dettagli, consulta [DisassociateRoleFromGroup AWS CLI Command Reference](#).

## **disassociate-service-role-from-account**

Il seguente esempio di codice mostra come utilizzare `disassociate-service-role-from-account`.

## AWS CLI

Per dissociare un ruolo di servizio dal proprio account AWS

L'`disassociate-service-role-from-account` seguente rimuove il ruolo di servizio associato all' AWS account. Se non si utilizza il ruolo di servizio in nessuna AWS regione, utilizzare il `delete-role-policy` comando per scollegare la politica `AWSGreengrassResourceAccessRolePolicy` gestita dal ruolo, quindi utilizzare il `delete-role` comando per eliminare il ruolo.

```
aws greengrass disassociate-service-role-from-account
```

Output:

```
{
  "DisassociatedAt": "2019-06-25T22:12:55Z"
}
```

Per ulteriori informazioni, consulta [Greengrass Service Role](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, vedere [DisassociateServiceRoleFromAccount](#) in AWS CLI Command Reference.

## get-associated-role

Il seguente esempio di codice mostra come utilizzare `get-associated-role`.

## AWS CLI

Per associare il ruolo a un gruppo Greengrass

L'`get-associated-role` seguente ottiene il IAM ruolo associato al gruppo Greengrass specificato. Il ruolo di gruppo viene utilizzato dalle funzioni e dai connettori Lambda locali per accedere ai servizi AWS .

```
aws greengrass get-associated-role \
  --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

Output:

```
{
  "RoleArn": "arn:aws:iam::123456789012:role/GG-Group-Role",
  "AssociatedAt": "2019-09-10T20:03:30Z"
}
```

Per ulteriori informazioni, consulta [Configurare il ruolo del gruppo](#) nella AWS IoT Greengrass Developer Guide.

- Per API i dettagli, consulta [GetAssociatedRole AWS CLI Command Reference](#).

## get-bulk-deployment-status

Il seguente esempio di codice mostra come utilizzare `get-bulk-deployment-status`.

### AWS CLI

Per verificare lo stato della distribuzione in blocco

L'esempio seguente recupera le informazioni sullo stato dell'operazione di distribuzione in blocco specificata. In questo esempio, il file che ha specificato i gruppi da distribuire ha un record di input non valido.

```
aws greengrass get-bulk-deployment-status \
  --bulk-deployment-id "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```

Output:

```
{
  "BulkDeploymentMetrics": {
    "InvalidInputRecords": 1,
    "RecordsProcessed": 1,
    "RetryAttempts": 0
  },
  "BulkDeploymentStatus": "Completed",
  "CreatedAt": "2019-06-25T16:11:33.265Z",
  "tags": {}
}
```

Per ulteriori informazioni, consulta [Create bulk deployments for Groups](#) nella IoT AWS Greengrass Developer Guide.

- Per i API dettagli, consulta [GetBulkDeploymentStatus Command Reference](#).AWS CLI

## get-connectivity-info

Il seguente esempio di codice mostra come utilizzare `get-connectivity-info`.

### AWS CLI

Per ottenere le informazioni di connettività per un core Greengrass

L'esempio seguente mostra gli endpoint che i dispositivi possono utilizzare per connettersi al core Greengrass specificato. Le informazioni sulla connettività sono un elenco di indirizzi IP o nomi di dominio, con i numeri di porta corrispondenti e metadati opzionali definiti dal cliente.

```
aws greengrass get-connectivity-info \  
  --thing-name "MyGroup_Core"
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

```

    }
  ]
}

```

- Per API i dettagli, vedere [GetConnectivityInfo](#) in AWS CLI Command Reference.

## get-connector-definition-version

Il seguente esempio di codice mostra come utilizzare `get-connector-definition-version`.

### AWS CLI

Per recuperare informazioni su una versione specifica della definizione di un connettore

L'esempio seguente recupera informazioni sulla versione specificata della definizione di connettore specificata. Per recuperare la versione IDs di tutte le versioni della definizione del connettore, utilizzate il `list-connector-definition-versions` comando. Per recuperare l'ID dell'ultima versione aggiunta alla definizione del connettore, utilizzate il `get-connector-definition` comando e controllate la `LatestVersion` proprietà.

```

aws greengrass get-connector-definition-version \
  --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8" \
  --connector-definition-version-id "63c57963-c7c2-4a26-a7e2-7bf478ea2623"

```

Output:

```

{
  "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"
}

```

Per ulteriori informazioni, consulta [Integrazione con servizi e protocolli utilizzando i connettori Greengrass](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetConnectorDefinitionVersionReference](#).

## get-connector-definition

Il seguente esempio di codice mostra come utilizzare `get-connector-definition`.

### AWS CLI

Per recuperare informazioni sulla definizione di un connettore

L'`get-connector-definition` esempio seguente recupera informazioni sulla definizione di connettore specificata. Per recuperare le definizioni IDs dei connettori, utilizzare il `list-connector-definitions` comando.

```

aws greengrass get-connector-definition \
  --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"

```

Output:

```

{
  "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": {}
}

```

Per ulteriori informazioni, consulta [Integrazione con servizi e protocolli utilizzando i connettori Greengrass](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetConnectorDefinition](#) Reference.

## get-core-definition-version

Il seguente esempio di codice mostra come utilizzare `get-core-definition-version`.

### AWS CLI

Per recuperare dettagli su una versione specifica della definizione di base di Greengrass

L'esempio seguente recupera informazioni sulla versione specificata della definizione principale specificata. Per recuperare tutte le versioni IDs della definizione principale, utilizzate il `list-core-definition-versions` comando. Per recuperare l'ID dell'ultima versione aggiunta alla definizione principale, usa il `get-core-definition` comando e controlla la `LatestVersion` proprietà.

```
aws greengrass get-core-definition-version \
  --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46" \
  --core-definition-version-id "42aeaac3-fd9d-4312-a8fd-ffa9404a20e0"
```

Output:

```
{
  "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": "42aeeac3-fd9d-4312-a8fd-ffa9404a20e0"
}
```

- Per API i dettagli, vedere [GetCoreDefinitionVersion](#) in AWS CLI Command Reference.

## get-core-definition

Il seguente esempio di codice mostra come utilizzare `get-core-definition`.

### AWS CLI

Per recuperare i dettagli per una definizione di base di Greengrass

L'`get-core-definition` esempio seguente recupera informazioni sulla definizione principale specificata. Per recuperare le IDs definizioni principali, utilizzate il `list-core-definitions` comando.

```
aws greengrass get-core-definition \
  --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46"
```

Output:

```
{
  "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": {}
}
```

- Per API i dettagli, vedere [GetCoreDefinition](#) in AWS CLI Command Reference.

## get-deployment-status

Il seguente esempio di codice mostra come utilizzare `get-deployment-status`.

## AWS CLI

Per recuperare lo stato di una distribuzione

L'`get-deployment-status` esempio seguente recupera lo stato per la distribuzione specificata del gruppo Greengrass specificato. Per ottenere l'ID di distribuzione, utilizzate il `list-deployments` comando e specificate l'ID del gruppo.

```
aws greengrass get-deployment-status \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731" \  
  --deployment-id "1065b8a0-812b-4f21-9d5d-e89b232a530f"
```

Output:

```
{  
  "DeploymentStatus": "Success",  
  "DeploymentType": "NewDeployment",  
  "UpdatedAt": "2019-06-18T17:04:44.761Z"  
}
```

- Per API i dettagli, vedere [GetDeploymentStatus](#) in AWS CLI Command Reference.

## get-device-definition-version

Il seguente esempio di codice mostra come utilizzare `get-device-definition-version`.

## AWS CLI

Per ottenere una versione di definizione del dispositivo

L'`get-device-definition-version` esempio seguente recupera informazioni sulla versione specificata della definizione del dispositivo specificata. Per recuperare la versione IDs di tutte le versioni della definizione del dispositivo, utilizzare il `list-device-definition-versions` comando. Per recuperare l'ID dell'ultima versione aggiunta alla definizione del dispositivo, utilizzate il `get-device-definition` comando e controllate la `LatestVersion` proprietà.

```
aws greengrass get-device-definition-version \  
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \  
  --device-definition-version-id "83c13984-6fed-447e-84d5-5b8aa45d5f71"
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [GetDeviceDefinitionVersion](#) in AWS CLI Command Reference.

## get-device-definition

Il seguente esempio di codice mostra come utilizzare `get-device-definition`.

### AWS CLI

Per ottenere una definizione di dispositivo

L'`get-device-definition` esempio seguente recupera le informazioni sulla definizione del dispositivo specificata. Per recuperare le definizioni IDs dei dispositivi, utilizzare il `list-device-definitions` comando.

```
aws greengrass get-device-definition \  
--device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

Output:

```
{  
  "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"  
}
```

- Per API i dettagli, vedere [GetDeviceDefinition](#) in AWS CLI Command Reference.

## get-function-definition-version

Il seguente esempio di codice mostra come utilizzare `get-function-definition-version`.

AWS CLI

Per recuperare dettagli su una versione specifica di una funzione Lambda

Di seguito `get-function-definition-version` vengono recuperate informazioni sulla versione specificata della definizione di funzione specificata. Per recuperare la versione IDs di tutte le versioni della definizione della funzione, utilizzate il `list-function-definition-versions` comando. Per recuperare l'ID dell'ultima versione aggiunta alla definizione della funzione, utilizzate il `get-function-definition` comando e controllate la `LatestVersion` proprietà.

```
aws greengrass get-function-definition-version \  
--function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85" \  
--function-definition-version-id "9748fda7-1589-4fcc-ac94-f5559e88678b"
```

Output:

```

{
  "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"
}

```

- Per API i dettagli, vedere [GetFunctionDefinitionVersion](#) in AWS CLI Command Reference.

## get-function-definition

Il seguente esempio di codice mostra come utilizzare `get-function-definition`.

## AWS CLI

Per recuperare la definizione di una funzione

L'`get-function-definition` seguente visualizza i dettagli per la definizione di funzione specificata. Per recuperare le definizioni IDs delle funzioni, utilizzare il `list-function-definitions` comando.

```
aws greengrass get-function-definition \  
  --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"
```

Output:

```
{  
  "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": {}  
}
```

- Per API i dettagli, vedere [GetFunctionDefinition](#) in AWS CLI Command Reference.

## get-group-certificate-authority

Il seguente esempio di codice mostra come utilizzare `get-group-certificate-authority`.

## AWS CLI

Per recuperare la CA associata a un gruppo Greengrass

L'`get-group-certificate-authority` seguente recupera l'autorità di certificazione (CA) associata al gruppo Greengrass specificato. Per ottenere l'ID dell'autorità di certificazione, utilizzate il `list-group-certificate-authorities` comando e specificate l'ID del gruppo.

```
aws greengrass get-group-certificate-authority \  
  --group-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"
```

```
--group-id "1013db12-8b58-45ff-acc7-704248f66731" \
--certificate-authority-
id "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6"
```

Output:

```
{
  "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/certificateauthorities/f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
  "GroupCertificateAuthorityId":
  "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
  "PemEncodedCertificate": "-----BEGIN CERTIFICATE-----
MIICiTCcAFICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBWEXAMPLEGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDEXAMPLEEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWf6
b24xFDASBgNVBAEXAMPLESDBb25zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWxhZAd
BgkqhkiG9w0BCQEWEG5vb25lQGftYXpvbi5jEXAMPLENMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0EXAMPLEBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWEXAMPLEDASBgNVBAwTC01BTSDBb25z
b2x1MRIwEAYDVQQDEwLUZXN0Q21sYWEXAMPLEGkqhkiG9w0BCQEWEG5vb25lQGft
YXpvbi5EXAMPLE8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySwTC2XADZ4nB+BLYgVIk60CEXAMPLE93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswYEXAMPLEEgpE
Ibb30hjZnzcvcQAARHhd1QWIMm2nrAgMBAAEwDQYJKEXAMPLEAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n"
}
```

- Per API i dettagli, vedere [GetGroupCertificateAuthority](#) in AWS CLI Command Reference.

## get-group-certificate-configuration

Il seguente esempio di codice mostra come utilizzare `get-group-certificate-configuration`.

AWS CLI

Per recuperare la configurazione per l'autorità di certificazione utilizzata dal gruppo Greengrass

L'`get-group-certificate-configuration` esempio seguente recupera la configurazione per l'autorità di certificazione (CA) utilizzata dal gruppo Greengrass specificato.

```
aws greengrass get-group-certificate-configuration \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Output:

```
{  
  "CertificateAuthorityExpiryInMilliseconds": 2524607999000,  
  "CertificateExpiryInMilliseconds": 604800000,  
  "GroupId": "1013db12-8b58-45ff-acc7-704248f66731"  
}
```

- Per API i dettagli, vedere [GetGroupCertificateConfiguration](#) in AWS CLI Command Reference.

## get-group-version

Il seguente esempio di codice mostra come utilizzare `get-group-version`.

AWS CLI

Per recuperare informazioni su una versione di un gruppo Greengrass

L'`get-group-version` esempio seguente recupera informazioni sulla versione specificata del gruppo specificato. Per recuperare la versione IDs di tutte le versioni del gruppo, utilizzate il `list-group-versions` comando. Per recuperare l'ID dell'ultima versione aggiunta al gruppo, usa il `get-group` comando e controlla la `LatestVersion` proprietà.

```
aws greengrass get-group-version \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731" \  
  --group-version-id "115136b3-cfd7-4462-b77f-8741a4b00e5e"
```

Output:

```
{  
  "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/42aeac3-  
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"
}

```

- Per API i dettagli, vedere [GetGroupVersion](#) in AWS CLI Command Reference.

## get-group

Il seguente esempio di codice mostra come utilizzare `get-group`.

### AWS CLI

Per recuperare informazioni su un gruppo Greengrass

L'`get-group` esempio seguente recupera le informazioni sul gruppo Greengrass specificato. Per recuperare i IDs tuoi gruppi, usa il comando. `list-groups`

```

aws greengrass get-group \
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"

```

Output:

```

{
  "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": {}
}

```

```
}
```

- Per API i dettagli, vedere [GetGroup](#) in AWS CLI Command Reference.

## get-logger-definition-version

Il seguente esempio di codice mostra come utilizzare `get-logger-definition-version`.

### AWS CLI

Per recuperare informazioni su una versione di una definizione di logger

L'`get-logger-definition-version` esempio seguente recupera informazioni sulla versione specificata della definizione di logger specificata. Per recuperare la versione IDs di tutte le versioni della definizione del logger, utilizzate il comando `list-logger-definition-versions`. Per recuperare l'ID dell'ultima versione aggiunta alla definizione del logger, usa il `get-logger-definition` comando e controlla la proprietà `LatestVersion`.

```
aws greengrass get-logger-definition-version \
  --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23" \
  --logger-definition-version-id "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [GetLoggerDefinitionVersion](#) in AWS CLI Command Reference.

## get-logger-definition

Il seguente esempio di codice mostra come utilizzare `get-logger-definition`.

## AWS CLI

Per recuperare informazioni sulla definizione di un logger

L'`get-logger-definition` seguente recupera informazioni sulla definizione del logger specificata. Per recuperare le definizioni IDs dei tuoi logger, usa il comando `list-logger-definitions`

```
aws greengrass get-logger-definition \  
  --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"
```

Output:

```
{  
  "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": {}  
}
```

- Per API i dettagli, vedere [GetLoggerDefinition](#) in AWS CLI Command Reference.

## get-resource-definition-version

Il seguente esempio di codice mostra come utilizzare `get-resource-definition-version`.

## AWS CLI

Per recuperare informazioni su una versione specifica di una definizione di risorsa

L'`get-resource-definition-version` seguente recupera informazioni sulla versione specificata della definizione di risorsa specificata. Per recuperare la versione IDs di tutte le versioni della definizione della risorsa, utilizzate il `list-resource-definition-versions` comando. Per recuperare l'ID dell'ultima versione aggiunta alla definizione della risorsa, usa il `get-resource-definition` comando e controlla la `LatestVersion` proprietà.

```
aws greengrass get-resource-definition-version \
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658" \
  --resource-definition-version-id "26e8829a-491a-464d-9c87-664bf6f6f2be"
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [GetResourceDefinitionVersion](#) in AWS CLI Command Reference.

## get-resource-definition

Il seguente esempio di codice mostra come utilizzare `get-resource-definition`.

### AWS CLI

Per recuperare informazioni sulla definizione di una risorsa

L'`get-resource-definition` esempio seguente recupera informazioni sulla definizione di risorsa specificata. Per recuperare le definizioni IDs delle risorse, utilizzare il `list-resource-definitions` comando.

```
aws greengrass get-resource-definition \  
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

Output:

```
{  
  "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": {}  
}
```

- Per API i dettagli, vedere [GetResourceDefinition](#) in AWS CLI Command Reference.

## get-service-role-for-account

Il seguente esempio di codice mostra come utilizzare `get-service-role-for-account`.

AWS CLI

Per recuperare i dettagli per il ruolo di servizio associato al tuo account

L'`get-service-role-for-account` esempio seguente recupera le informazioni sul ruolo di servizio associato all'account AWS .

```
aws greengrass get-service-role-for-account
```

Output:

```
{  
  "AssociatedAt": "2018-10-18T15:59:20Z",
```

```

    "RoleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
  }

```

Per ulteriori informazioni, consulta [Greengrass Service Role](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, vedere [GetServiceRoleForAccount](#) in AWS CLI Command Reference.

## get-subscription-definition-version

Il seguente esempio di codice mostra come utilizzare `get-subscription-definition-version`.

### AWS CLI

Per recuperare informazioni su una versione specifica di una definizione di abbonamento

L'`get-subscription-definition-version` esempio seguente recupera e recupera le informazioni sulla versione specificata della definizione di sottoscrizione specificata. Per recuperare la versione IDs di tutte le versioni della definizione di sottoscrizione, utilizzate il comando `list-subscription-definition-versions`. Per recuperare l'ID dell'ultima versione aggiunta alla definizione di abbonamento, usa il `get-subscription-definition` comando e controlla la `LatestVersion` proprietà.

```

aws greengrass get-subscription-definition-version \
  --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152" \
  --subscription-definition-version-id "88ae8699-12ac-4663-ba3f-4d7f0519140b"

```

Output:

```

{
  "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"
}

```

- Per API i dettagli, vedere [GetSubscriptionDefinitionVersion](#) in AWS CLI Command Reference.

## get-subscription-definition

Il seguente esempio di codice mostra come utilizzare `get-subscription-definition`.

### AWS CLI

Per recuperare informazioni sulla definizione di una sottoscrizione

L'`get-subscription-definition` esempio seguente recupera informazioni sulla definizione di sottoscrizione specificata. Per recuperare le definizioni IDs delle sottoscrizioni, utilizzare il `list-subscription-definitions` comando.

```

aws greengrass get-subscription-definition \
  --subscription-definition-id 70e49321-83d5-45d2-bc09-81f4917ae152

```

Output:

```

{
  "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": {}
}

```

- Per API i dettagli, vedere [GetSubscriptionDefinition](#) in AWS CLI Command Reference.

## get-thing-runtime-configuration

Il seguente esempio di codice mostra come utilizzare `get-thing-runtime-configuration`.

### AWS CLI

Per recuperare la configurazione di runtime di un core Greengrass

L'esempio seguente recupera la configurazione di runtime di un core Greengrass. Prima di poter recuperare la configurazione di runtime, è necessario utilizzare il `update-thing-runtime-configuration` comando per creare una configurazione di runtime per il core.

```
aws greengrass get-thing-runtime-configuration \  
  --thing-name SampleGreengrassCore
```

Output:

```
{  
  "RuntimeConfiguration": {  
    "TelemetryConfiguration": {  
      "ConfigurationSyncStatus": "OutOfSync",  
      "Telemetry": "On"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Configurazione delle impostazioni di telemetria nella IoT AWS Greengrass Developer Guide](#).

- Per i API dettagli, vedere in Command Reference. [GetThingRuntimeConfiguration](#) AWS CLI

## list-bulk-deployment-detailed-reports

Il seguente esempio di codice mostra come utilizzare `list-bulk-deployment-detailed-reports`.

### AWS CLI

Per elencare informazioni sulle singole distribuzioni in una distribuzione collettiva



L'`list-bulk-deployment-detailed-report` seguente visualizza informazioni sulle singole distribuzioni in un'operazione di distribuzione in blocco, incluso lo stato.

```
aws greengrass list-bulk-deployment-detailed-reports \  
  --bulk-deployment-id 42ce9c42-489b-4ed4-b905-8996aa50ef9d
```

Output:

```
{  
  "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"  
    },  
    ...  
  ]  
}
```

Per ulteriori informazioni, consulta [Create bulk deployments for Groups](#) nella IoT AWS Greengrass Developer Guide.

- Per i API dettagli, consulta [ListBulkDeploymentDetailedReports](#) Command Reference.AWS CLI

## list-bulk-deployments

Il seguente esempio di codice mostra come utilizzare `list-bulk-deployments`.

### AWS CLI

Per elencare le distribuzioni in blocco

L'`list-bulk-deployments` esempio seguente elenca tutte le distribuzioni in blocco.

```
aws greengrass list-bulk-deployments
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Create bulk deployments for Groups](#) nella IoT AWS Greengrass Developer Guide.

- Per i API dettagli, consulta [ListBulkDeployments](#) Command Reference.AWS CLI

## list-connector-definition-versions

Il seguente esempio di codice mostra come utilizzare `list-connector-definition-versions`.

### AWS CLI

Per elencare le versioni disponibili per la definizione di un connettore

L'`list-connector-definition-versions` esempio seguente elenca le versioni disponibili per la definizione di connettore specificata. Utilizzate il `list-connector-definitions` comando per ottenere l'ID della definizione del connettore.

```
aws greengrass list-connector-definition-versions \  
--connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Integrazione con servizi e protocolli utilizzando i connettori Greengrass](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListConnectorDefinitionVersionsReference](#).

## list-connector-definitions

Il seguente esempio di codice mostra come utilizzare `list-connector-definitions`.

AWS CLI

Per elencare i connettori Greengrass definiti

L'`list-connector-definitions` esempio seguente elenca tutti i connettori Greengrass definiti per il tuo AWS account.

```
aws greengrass list-connector-definitions
```

Output:

```
{  
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Integrazione con servizi e protocolli utilizzando i connettori Greengrass](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListConnectorDefinitionsReference](#).

## list-core-definition-versions

Il seguente esempio di codice mostra come utilizzare `list-core-definition-versions`.

### AWS CLI

Per elencare le versioni di una definizione di base di Greengrass

L'`list-core-definitions` esempio seguente elenca tutte le versioni della definizione principale della specie Greengrass. È possibile utilizzare il `list-core-definitions` comando per ottenere l'ID della versione.

```

aws greengrass list-core-definition-versions \
  --core-definition-id "eaf280cb-138c-4d15-af36-6f681a1348f7"

```

Output:

```

{
  "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"
    }
]
}

```

- Per API i dettagli, vedere [ListCoreDefinitionVersions](#) in AWS CLI Command Reference.

## list-core-definitions

Il seguente esempio di codice mostra come utilizzare `list-core-definitions`.

### AWS CLI

Per elencare le definizioni principali di Greengrass

L'`list-core-definitions` esempio seguente elenca tutte le definizioni principali di Greengrass per il tuo AWS account.

```
aws greengrass list-core-definitions
```

Output:

```

{
  "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"
    }
]
}

```

- Per API i dettagli, vedere [ListCoreDefinitions](#) in AWS CLI Command Reference.

## list-deployments

Il seguente esempio di codice mostra come utilizzare `list-deployments`.

### AWS CLI

Per elencare le distribuzioni per un gruppo Greengrass

L'`list-deployments` esempio seguente elenca le distribuzioni per il gruppo Greengrass specificato. È possibile utilizzare il `list-groups` comando per cercare l'ID del gruppo.

```
aws greengrass list-deployments \  
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [ListDeployments](#) in AWS CLI Command Reference.

## list-device-definition-versions

Il seguente esempio di codice mostra come utilizzare `list-device-definition-versions`.

AWS CLI

Per elencare le versioni di una definizione di dispositivo

L'`list-device-definition-versions` esempio seguente visualizza le versioni della definizione del dispositivo associate alla definizione del dispositivo specificata.

```
aws greengrass list-device-definition-versions \  
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

Output:

```
{  
  "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"
    }
]
}

```

- Per API i dettagli, vedere [ListDeviceDefinitionVersions](#) in AWS CLI Command Reference.

## list-device-definitions

Il seguente esempio di codice mostra come utilizzare `list-device-definitions`.

### AWS CLI

Per elencare le definizioni dei dispositivi

L'`list-device-definitions` esempio seguente mostra i dettagli sulle definizioni dei dispositivi nel tuo AWS account nella AWS regione specificata.

```

aws greengrass list-device-definitions \
  --region us-west-2

```

Output:

```

{
  "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"
  }
]
}

```

- Per API i dettagli, vedere [ListDeviceDefinitions](#) in AWS CLI Command Reference.

## list-function-definition-versions

Il seguente esempio di codice mostra come utilizzare `list-function-definition-versions`.

### AWS CLI

Per elencare le versioni di una funzione Lambda

L'`list-function-definition-versions` seguente elenca tutte le versioni della funzione Lambda specificata. È possibile utilizzare il `list-function-definitions` comando per ottenere l'ID.

```
aws greengrass list-function-definition-versions \  
--function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

```
]
}
```

- Per API i dettagli, vedere [ListFunctionDefinitionVersions](#) in AWS CLI Command Reference.

## list-function-definitions

Il seguente esempio di codice mostra come utilizzare `list-function-definitions`.

### AWS CLI

Per elencare le funzioni Lambda

L'`list-function-definitions` esempio seguente elenca tutte le funzioni Lambda definite per il tuo AWS account.

```
aws greengrass list-function-definitions
```

Output:

```
{
  "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"
  }
]
}

```

- Per API i dettagli, consulta [ListFunctionDefinitions AWS CLI Command Reference](#).

## list-group-certificate-authorities

Il seguente esempio di codice mostra come utilizzare `list-group-certificate-authorities`.

### AWS CLI

Per elencare la corrente CAs di un gruppo

L'`list-group-certificate-authorities` esempio seguente elenca le autorità di certificazione correnti (CAs) per il gruppo Greengrass specificato.

```
aws greengrass list-group-certificate-authorities \
```

```
--group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Output:

```
{
  "GroupCertificateAuthorities": [
    {
      "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-
west-2:123456789012:/greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/
certificateauthorities/
f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
      "GroupCertificateAuthorityId":
"f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6"
    }
  ]
}
```

- Per API i dettagli, vedere [ListGroupCertificateAuthorities](#) in AWS CLI Command Reference.

## list-group-versions

Il seguente esempio di codice mostra come utilizzare `list-group-versions`.

AWS CLI

Per elencare le versioni di un gruppo Greengrass

L'`list-group-versions` esempio seguente elenca le versioni del gruppo Greengrass specificato.

```
aws greengrass list-group-versions \
  --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

Output:

```
{
  "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"
    }
]
}

```

- Per API i dettagli, vedere [ListGroupVersions](#) in AWS CLI Command Reference.

## list-groups

Il seguente esempio di codice mostra come utilizzare `list-groups`.

## AWS CLI

Per elencare i gruppi Greengrass

L'`list-groups` seguente elenca tutti i gruppi Greengrass definiti nel tuo AWS account.

```
aws greengrass list-groups
```

Output:

```
{
  "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"
  }
]
}

```

- Per API i dettagli, vedere [ListGroups](#) in AWS CLI Command Reference.

## list-logger-definition-versions

Il seguente esempio di codice mostra come utilizzare `list-logger-definition-versions`.

### AWS CLI

Per ottenere un elenco di versioni di una definizione di logger

L'`list-logger-definition-versions`esempio seguente ottiene un elenco di tutte le versioni della definizione di logger specificata.

```

aws greengrass list-logger-definition-versions \
  --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"

```

Output:

```

{
  "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"
    }
]
}

```

- Per API i dettagli, vedere [ListLoggerDefinitionVersions](#) in AWS CLI Command Reference.

## list-logger-definitions

Il seguente esempio di codice mostra come utilizzare `list-logger-definitions`.

### AWS CLI

Per ottenere un elenco di definizioni dei logger

L'`list-logger-definitions` esempio seguente elenca tutte le definizioni dei logger per il tuo AWS account.

```
aws greengrass list-logger-definitions
```

Output:

```

{
  "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"
    }
  ]
}

```

- Per API i dettagli, vedere [ListLoggerDefinitions](#) in AWS CLI Command Reference.

## list-resource-definition-versions

Il seguente esempio di codice mostra come utilizzare `list-resource-definition-versions`.

### AWS CLI

Per elencare le versioni di una definizione di risorsa

L'`list-resource-definition-versions` seguente elenca le versioni per la risorsa Greengrass specificata.

```
aws greengrass list-resource-definition-versions \
  --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ListResourceDefinitionVersions](#) in AWS CLI Command Reference.

## list-resource-definitions

Il seguente esempio di codice mostra come utilizzare `list-resource-definitions`.

## AWS CLI

Per elencare le risorse che sono definite

L'`list-resource-definitions`esempio seguente elenca le risorse definite per l'utilizzo di AWS IoT Greengrass.

```
aws greengrass list-resource-definitions
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ListResourceDefinitions](#) in AWS CLI Command Reference.

## list-subscription-definition-versions

Il seguente esempio di codice mostra come utilizzare `list-subscription-definition-versions`.

### AWS CLI

Per elencare le versioni di una definizione di abbonamento

L'`list-subscription-definition-versions` seguente elenca tutte le versioni dell'abbonamento specificato. È possibile utilizzare il `list-subscription-definitions` comando per cercare l'ID dell'abbonamento.

```
aws greengrass list-subscription-definition-versions \  
--subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152"
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [ListSubscriptionDefinitionVersions](#) in AWS CLI Command Reference.

## list-subscription-definitions

Il seguente esempio di codice mostra come utilizzare `list-subscription-definitions`.

### AWS CLI

Per ottenere una lista, sottoscrivere le definizioni

L'`list-subscription-definitions` seguente elenca tutti gli abbonamenti AWS IoT Greengrass definiti nel tuo account. AWS

```
aws greengrass list-subscription-definitions
```

Output:

```
{
  "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"
    }
]
}

```

- Per API i dettagli, vedere [ListSubscriptionDefinitions](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag allegati a una risorsa

L'`list-tags-for-resource` esempio seguente elenca i tag e i relativi valori associati alla risorsa specificata.

```

aws greengrass list-tags-for-resource \
  --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658"

```

Output:

```

{
  "tags": {
    "ResourceSubType": "USB",
    "ResourceType": "Device"
  }
}

```

Per ulteriori informazioni, consulta [Tagging Your Greengrass Resources nella AWS IoT Greengrass Developer Guide](#).

- Per API i dettagli, consulta Command [ListTagsForResource](#) Reference AWS CLI .

## reset-deployments

Il seguente esempio di codice mostra come utilizzare `reset-deployments`.

### AWS CLI

Per ripulire le informazioni di distribuzione per un gruppo Greengrass

L'`reset-deployments` esempio seguente ripulisce le informazioni di distribuzione per il gruppo Greengrass specificato. Quando si aggiunge il `--force` option, le informazioni di distribuzione vengono reimpostate senza attendere che il dispositivo principale risponda.

```
aws greengrass reset-deployments \  
  --group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \  
  --force
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Reset Deployments](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, vedere [ResetDeployments](#) in AWS CLI Command Reference.

## start-bulk-deployment

Il seguente esempio di codice mostra come utilizzare `start-bulk-deployment`.

### AWS CLI

Per avviare un'operazione di distribuzione in blocco

L'`start-bulk-deployment` esempio seguente avvia un'operazione di distribuzione in blocco, utilizzando un file archiviato in un bucket S3 per specificare i gruppi da distribuire.

```
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\"}"
```

Output:

```
{  
  "BulkDeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
bulk/deployments/870fb41b-6288-4e0c-bc76-a7ba4b4d3267",  
  "BulkDeploymentId": "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"  
}
```

Per ulteriori informazioni, consulta [Create bulk deployments for Groups](#) nella IoT AWS Greengrass Developer Guide.

- Per i API dettagli, consulta [StartBulkDeployment](#) Command Reference.AWS CLI

## stop-bulk-deployment

Il seguente esempio di codice mostra come utilizzare `stop-bulk-deployment`.

AWS CLI

Per interrompere una distribuzione di massa

L'esempio seguente interrompe la distribuzione in blocco specificata. Se tenti di interrompere una distribuzione di massa completa, ricevi un errore: `InvalidInputException: Cannot change state of finished execution.`

```
aws greengrass stop-bulk-deployment \  
  --bulk-deployment-id "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Create bulk deployments for Groups](#) nella IoT AWS Greengrass Developer Guide.

- Per i API dettagli, consulta [StopBulkDeployment](#) Command Reference.AWS CLI



## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per applicare tag a una risorsa

L'esempio seguente applica due tag `ResourceType` e `ResourceSubType` alla risorsa Greengrass specificata. Questa operazione può sia aggiungere nuovi tag e valori sia aggiornare il valore dei tag esistenti. Usa il `untag-resource` comando per rimuovere un tag.

```
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"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Your Greengrass Resources nella AWS IoT Greengrass Developer Guide](#).

- Per API i dettagli, consulta Command [TagResource](#) Reference AWS CLI .

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag e il relativo valore da una risorsa

L'esempio seguente rimuove il tag la cui chiave proviene `Category` dal gruppo Greengrass specificato. Se la chiave `Category` non esiste per la risorsa specificata, non viene restituito alcun errore.

```
aws greengrass untag-resource \  
  --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
groups/1013db12-8b58-45ff-acc7-704248f66731" \  
  --tag-keys "Category"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Your Greengrass Resources nella AWS IoT Greengrass Developer Guide](#).

- Per API i dettagli, consulta Command [UntagResource](#)Reference AWS CLI .

## update-connectivity-info

Il seguente esempio di codice mostra come utilizzare `update-connectivity-info`.

### AWS CLI

Per aggiornare le informazioni di connettività per un core Greengrass

L'`update-connectivity-info`esempio seguente modifica gli endpoint che i dispositivi possono utilizzare per connettersi al core Greengrass specificato. Le informazioni sulla connettività sono un elenco di indirizzi IP o nomi di dominio, con i numeri di porta corrispondenti e metadati opzionali definiti dal cliente. Potrebbe essere necessario aggiornare le informazioni di connettività quando la rete locale cambia.

```
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"}]"
```

Output:

```
{  
  "Version": "312de337-59af-4cf9-a278-2a23bd39c300"  
}
```

- Per API i dettagli, vedere [UpdateConnectivityInfo](#)in AWS CLI Command Reference.

## update-connector-definition

Il seguente esempio di codice mostra come utilizzare `update-connector-definition`.

### AWS CLI

Per aggiornare il nome di una definizione di connettore

L'update-connector-definitionesempio seguente aggiorna il nome della definizione di connettore specificata. Se desideri aggiornare i dettagli del connettore, usa il create-connector-definition-version comando per creare una nuova versione.

```
aws greengrass update-connector-definition \  
  --connector-definition-id "55d0052b-0d7d-44d6-b56f-21867215e118" \  
  --name "GreengrassConnectors2019"
```

Per ulteriori informazioni, consulta [Integrazione con servizi e protocolli tramite connettori](#) nella AWS IoT Greengrass Developer Guide.

- Per API i dettagli, vedere [UpdateConnectorDefinition](#) in AWS CLI Command Reference.

## update-core-definition

Il seguente esempio di codice mostra come utilizzare update-core-definition.

### AWS CLI

Per aggiornare una definizione di base

L'update-core-definitionesempio seguente modifica il nome della definizione principale specificata. È possibile aggiornare solo la name proprietà di una definizione principale.

```
aws greengrass update-core-definition \  
  --core-definition-id "582efe12-b05a-409e-9a24-a2ba1bcc4a12" \  
  --name "MyCoreDevices"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configurare AWS IoT Greengrass Core nella IoT AWS Greengrass](#) Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateCoreDefinition](#) Reference.

## update-device-definition

Il seguente esempio di codice mostra come utilizzare update-device-definition.

### AWS CLI

Per aggiornare la definizione di un dispositivo

L'update-device-definitionesempio seguente modifica il nome della definizione del dispositivo specificata. È possibile aggiornare solo la name proprietà di una definizione di dispositivo.

```
aws greengrass update-device-definition \  
  --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \  
  --name "TemperatureSensors"
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [UpdateDeviceDefinition](#) in AWS CLI Command Reference.

## update-function-definition

Il seguente esempio di codice mostra come utilizzareupdate-function-definition.

### AWS CLI

Per aggiornare il nome di una definizione di funzione

L'update-function-definitionesempio seguente aggiorna il nome della definizione di funzione specificata. Se desiderate aggiornare i dettagli della funzione, utilizzate il create-function-definition-version comando per creare una nuova versione.

```
aws greengrass update-function-definition \  
  --function-definition-id "e47952bd-dea9-4e2c-a7e1-37bbe8807f46" \  
  --name ObsoleteFunction
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Run Local Lambda Functions](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateFunctionDefinition](#)Reference.

## update-group-certificate-configuration

Il seguente esempio di codice mostra come utilizzareupdate-group-certificate-configuration.

## AWS CLI

Per aggiornare la scadenza dei certificati di un gruppo

L'`update-group-certificate-configuration` seguente imposta una scadenza di 10 giorni per i certificati generati per il gruppo specificato.

```
aws greengrass update-group-certificate-configuration \  
  --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1" \  
  --certificate-expiry-in-milliseconds 864000000
```

Output:

```
{  
  "CertificateExpiryInMilliseconds": 864000000,  
  "CertificateAuthorityExpiryInMilliseconds": 2524607999000,  
  "GroupId": "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"  
}
```

Per ulteriori informazioni, consulta [AWS IoT Greengrass Security nella IoT AWS Greengrass Developer Guide](#).

- Per API i dettagli, consulta [AWS CLI Command UpdateGroupCertificateConfiguration Reference](#).

## update-group

Il seguente esempio di codice mostra come utilizzare `update-group`.

### AWS CLI

Per aggiornare il nome del gruppo

L'`update-group` seguente aggiorna il nome del gruppo Greengrass specificato. Se desideri aggiornare i dettagli del gruppo, usa il `create-group-version` comando per creare una nuova versione.

```
aws greengrass update-group \  
  --group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \  
  --name TestGroup4of6
```

Per ulteriori informazioni, consulta [Configure AWS IoT Greengrass on IoT nella AWS IoT AWS Greengrass Developer Guide](#).

- Per API i dettagli, consulta AWS CLI Command [UpdateGroup](#)Reference.

## update-logger-definition

Il seguente esempio di codice mostra come utilizzare `update-logger-definition`.

### AWS CLI

Per aggiornare la definizione di un logger

L'`update-logger-definition`esempio seguente modifica il nome della definizione del logger specificata. È possibile aggiornare solo la name proprietà di una definizione di logger.

```
aws greengrass update-logger-definition \  
  --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0" \  
  --name "LoggingConfigsForSensors"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Monitoring with AWS IoT Greengrass Logs](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, vedere [UpdateLoggerDefinition](#)in AWS CLI Command Reference.

## update-resource-definition

Il seguente esempio di codice mostra come utilizzare `update-resource-definition`.

### AWS CLI

Per aggiornare il nome di una definizione di risorsa

L'`update-resource-definition`esempio seguente aggiorna il nome della definizione di risorsa specificata. Se desideri modificare i dettagli della risorsa, usa il `create-resource-definition-version` comando per creare una nuova versione.

```
aws greengrass update-resource-definition \  
  --resource-definition-id "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38" \  
  --name GreengrassConnectorResources
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Access Local Resources with Lambda Functions and Connectors](#) nella IoT AWS Greengrass Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateResourceDefinition](#) Reference.

## update-subscription-definition

Il seguente esempio di codice mostra come utilizzare `update-subscription-definition`.

### AWS CLI

Per aggiornare il nome di una definizione di abbonamento

L'`update-subscription-definition` esempio seguente aggiorna il nome per la definizione di sottoscrizione specificata. Se desideri modificare i dettagli dell'abbonamento, usa il `create-subscription-definition-version` comando per creare una nuova versione.

```
aws greengrass update-subscription-definition \  
  --subscription-definition-id "fa81bc84-3f59-4377-a84b-5d0134da359b" \  
  --name "ObsoleteSubscription"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta il titolo della guida.

- Per API i dettagli, vedere [UpdateSubscriptionDefinition](#) in AWS CLI Command Reference.

## update-thing-runtime-configuration

Il seguente esempio di codice mostra come utilizzare `update-thing-runtime-configuration`.

### AWS CLI

Per attivare la telemetria nella configurazione di runtime di un core Greengrass

L'`update-thing-runtime-configuration` esempio seguente aggiorna la configurazione di runtime di un core Greengrass per attivare la telemetria.

```
aws greengrass update-thing-runtime-configuration \  
  --thing-name SampleGreengrassCore \  
  --runtime-configuration-name TelemetryConfiguration
```

```
--telemetry-configuration {"Telemetry\\":\\"0n\\"}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configurazione delle impostazioni di telemetria nella IoT AWS Greengrass Developer Guide](#).

- Per i API dettagli, vedere in Command Reference. [UpdateThingRuntimeConfiguration](#) AWS CLI

## AWS IoT Greengrass V2 esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS IoT Greengrass V2.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **associate-service-role-to-account**

Il seguente esempio di codice mostra come utilizzare `associate-service-role-to-account`.

AWS CLI

Per associare il ruolo di servizio Greengrass al tuo account AWS

L'esempio seguente associa un ruolo di servizio a AWS IoT Greengrass per AWS il tuo account.

```
aws greengrassv2 associate-service-role-to-account \  
--role-arn arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole
```

Output:



```
{
  "associatedAt": "2022-01-19T19:21:53Z"
}
```

Per ulteriori informazioni, consulta il [ruolo del servizio Greengrass nella Guida](#) per sviluppatori IoT AWS Greengrass V2.

- Per API i dettagli, vedere [AssociateServiceRoleToAccount](#) in AWS CLI Command Reference.

## batch-associate-client-device-with-core-device

Il seguente esempio di codice mostra come utilizzare `batch-associate-client-device-with-core-device`.

### AWS CLI

Per associare i dispositivi client a un dispositivo principale

L'esempio seguente associa due dispositivi client a un dispositivo principale.

```
aws greengrassv2 batch-associate-client-device-with-core-device \
  --core-device-thing-name MyGreengrassCore \
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

Output:

```
{
  "errorEntries": []
}
```

Per ulteriori informazioni, consulta [Interagire con i dispositivi IoT locali](#) nella Guida per AWS sviluppatori IoT Greengrass V2.

- Per API i dettagli, vedere [BatchAssociateClientDeviceWithCoreDevice](#) in AWS CLI Command Reference.

## batch-disassociate-client-device-from-core-device

Il seguente esempio di codice mostra come utilizzare `batch-disassociate-client-device-from-core-device`.

## AWS CLI

Per dissociare i dispositivi client da un dispositivo principale

L'`batch-disassociate-client-device-from-core-device`esempio seguente dissocia due dispositivi client da un dispositivo principale.

```
aws greengrassv2 batch-disassociate-client-device-from-core-device \  
  --core-device-thing-name MyGreengrassCore \  
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

Output:

```
{  
  "errorEntries": []  
}
```

Per ulteriori informazioni, consulta [Interagire con i dispositivi IoT locali](#) nella Guida per AWS sviluppatori IoT Greengrass V2.

- Per API i dettagli, vedere [BatchDisassociateClientDeviceFromCoreDevice](#)in AWS CLI Command Reference.

## cancel-deployment

Il seguente esempio di codice mostra come utilizzare`cancel-deployment`.

### AWS CLI

Per annullare una distribuzione

L'`cancel-deployment`esempio seguente interrompe una distribuzione continua in un gruppo di oggetti.

```
aws greengrassv2 cancel-deployment \  
  --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "message": "SUCCESS"
```

```
}
```

Per ulteriori informazioni, consulta [Annullare le implementazioni](#) nella Guida per gli sviluppatori di IoT AWS Greengrass V2.

- Per i API dettagli, vedere [CancelDeployment](#) in Command Reference.AWS CLI

## create-component-version

Il seguente esempio di codice mostra come utilizzare `create-component-version`.

### AWS CLI

Esempio 1: creare una versione di un componente da una ricetta

L'`create-component-version`esempio seguente crea una versione di un componente Hello World da un file di ricette.

```
aws greengrassv2 create-component-version \  
  --inline-recipe fileb://com.example.HelloWorld-1.0.0.json
```

Contenuto di `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}'"  
      }  
    }  
  ]  
}
```

```

    }
  ]
}

```

### Output:

```

{
  "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": {}
  }
}

```

Per ulteriori informazioni, consulta [Creare componenti personalizzati](#) e [Caricare componenti da distribuire nella Guida](#) per sviluppatori AWS IoT Greengrass V2.

### Esempio 2: creare una versione di un componente da una funzione AWS Lambda

L'`create-component-version` seguente crea una versione di un componente Hello World da una funzione AWS Lambda.

```

aws greengrassv2 create-component-version \
  --cli-input-json file://lambda-function-component.json

```

Contenuto di `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"
  }
]
}
}

```

Output:

```

{
  "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": {}
  }
}

```

Per ulteriori informazioni, consulta [Run AWS Lambda functions](#) nella IoT AWS Greengrass V2 Developer Guide.

- Per API i dettagli, consulta Command [CreateComponentVersion](#) Reference AWS CLI .

## create-deployment

Il seguente esempio di codice mostra come utilizzare `create-deployment`.

### AWS CLI

Esempio 1: creare una distribuzione

L'`create-deployment` esempio seguente distribuisce l'interfaccia a riga di comando AWS IoT Greengrass su un dispositivo principale.

```

aws greengrassv2 create-deployment \
  --cli-input-json file://cli-deployment.json

```

Contenuto di `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": {}
}
```

#### Output:

```
{
  "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Per ulteriori informazioni, consulta [Create deployments](#) nella IoT AWS Greengrass V2 Developer Guide.

#### Esempio 2: creare una distribuzione che aggiorni le configurazioni dei componenti

L'create-deploymentesempio seguente distribuisce il componente AWS IoT Greengrass nucleus su un gruppo di dispositivi principali. Questa distribuzione applica i seguenti aggiornamenti di configurazione per il componente nucleus:

Reimposta le impostazioni proxy dei dispositivi di destinazione ai valori predefiniti senza impostazioni proxy. Reimposta le impostazioni dei dispositivi di destinazione ai valori predefiniti. Imposta MQTT le opzioni per il nucleo'.Imposta il livello di registrazione per il JVM nucleo. JVM

```
aws greengrassv2 create-deployment \
```

```
--cli-input-json file://nucleus-deployment.json
```

Contenuto di 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
\\\"}}\"
      }
    }
  },
  "deploymentPolicies": {
    "failureHandlingPolicy": "ROLLBACK",
    "componentUpdatePolicy": {
      "timeoutInSeconds": 60,
      "action": "NOTIFY_COMPONENTS"
    },
    "configurationValidationPolicy": {
      "timeoutInSeconds": 60
    }
  },
  "iotJobConfiguration": {}
}
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Creare implementazioni e aggiornare le configurazioni dei componenti](#) nella Guida per sviluppatori IoT AWS Greengrass V2.

- Per i API dettagli, vedere in Command Reference. [CreateDeployment](#) AWS CLI

## delete-component

Il seguente esempio di codice mostra come utilizzare `delete-component`.

### AWS CLI

Per eliminare la versione di un componente

L'`delete-component` esempio seguente elimina un componente Hello World.

```
aws greengrassv2 delete-component \  
  --arn arn:aws:greengrass:us-west-2:123456789012:components:com.example>HelloWorld:versions:1.0.0
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Manage components](#) nella AWS IoT Greengrass V2 Developer Guide.

- Per API i dettagli, vedere [DeleteComponent](#) in AWS CLI Command Reference.

## delete-core-device

Il seguente esempio di codice mostra come utilizzare `delete-core-device`.

### AWS CLI

Per eliminare un dispositivo principale

L'`delete-core-device` esempio seguente elimina un dispositivo principale AWS IoT Greengrass.

```
aws greengrassv2 delete-core-device \  
  --core-device-thing-name MyGreengrassCore
```

Questo comando non produce alcun output.



Per ulteriori informazioni, consulta [Disinstallare il software AWS IoT Greengrass Core](#) nella Guida per sviluppatori IoT AWS Greengrass V2.

- Per API i dettagli, vedere [DeleteCoreDevice](#) in AWS CLI Command Reference.

## describe-component

Il seguente esempio di codice mostra come utilizzare `describe-component`.

### AWS CLI

Per descrivere la versione di un componente

L'`describe-component` esempio seguente descrive un componente Hello World.

```
aws greengrassv2 describe-component \  
  --arn arn:aws:greengrass:us-  
west-2:123456789012:components:com.example>HelloWorld:versions:1.0.0
```

Output:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Manage components](#) nella AWS IoT Greengrass V2 Developer Guide.

- Per API i dettagli, vedere [DescribeComponent](#) in AWS CLI Command Reference.

## **disassociate-service-role-from-account**

Il seguente esempio di codice mostra come utilizzare `disassociate-service-role-from-account`.

### AWS CLI

Per dissociare il ruolo di servizio Greengrass dal tuo account AWS

L'esempio seguente dissocia il ruolo del servizio Greengrass da IoT Greengrass per il tuo account AWS.

```
aws greengrassv2 disassociate-service-role-from-account
```

Output:

```
{
  "disassociatedAt": "2022-01-19T19:26:09Z"
}
```

Per ulteriori informazioni, consulta il [ruolo del servizio Greengrass nella Guida](#) per sviluppatori IoT AWS Greengrass V2.

- Per API i dettagli, vedere [DisassociateServiceRoleFromAccount](#) in AWS CLI Command Reference.

## **get-component-version-artifact**

Il seguente esempio di codice mostra come utilizzare `get-component-version-artifact`.

### AWS CLI

Per URL scaricare un elemento componente

L'esempio seguente consente di URL scaricare il file del componente della console di debug locale. 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"
```

Output:

```
{
  "preSignedUrl": "https://evergreencomponentmanageme-
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..."
}
```

Per ulteriori informazioni, consulta [Manage components](#) nella AWS IoT Greengrass V2 Developer Guide.

- Per API i dettagli, vedere [GetComponentVersionArtifact](#) in AWS CLI Command Reference.

## get-component

Il seguente esempio di codice mostra come utilizzare `get-component`.

### AWS CLI

Esempio 1: scaricare la ricetta di un componente in YAML formato (Linux, macOS o Unix)

L'`get-component` esempio seguente scarica la ricetta di un componente Hello World in un file in YAML formato. Questo comando funziona nel modo seguente:

Utilizza i `--query` parametri `--output` and per controllare l'output del comando. Questi parametri estraggono il blob della ricetta dall'output del comando. Per ulteriori informazioni sul controllo dell'output, vedere [Controlling Command Output nella AWS Command Line Interface User Guide](#). Utilizza l'`base64` utilità. Questa utilità decodifica il blob estratto nel testo originale. Il blob restituito da un `get-component` comando riuscito è testo con codifica base64. È necessario decodificare questo blob per ottenere il testo originale. Salva il testo decodificato in un file. La sezione finale del comando (`> com.example.HelloWorld-1.0.0.json`) salva il testo decodificato in un file.

```
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
```

Per ulteriori informazioni, consulta [Manage components](#) nella AWS IoT Greengrass V2 Developer Guide.

Esempio 2: Per scaricare la ricetta di un componente in YAML formato (Windows) CMD

L'get-componentesempio seguente scarica la ricetta di un componente Hello World in un file in YAML formato. Questo comando utilizza l'certutilutilità.

```
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
```

Per ulteriori informazioni, consulta [Manage components](#) nella AWS IoT Greengrass V2 Developer Guide.

Esempio 3: Per scaricare la ricetta di un componente in YAML formato (Windows) PowerShell

L'get-componentesempio seguente scarica la ricetta di un componente Hello World in un file in YAML formato. Questo comando utilizza l'certutilutilità.

```
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
```

Per ulteriori informazioni, consulta [Manage components](#) nella AWS IoT Greengrass V2 Developer Guide.

- Per API i dettagli, vedere [GetComponent](#) in AWS CLI Command Reference.

## get-connectivity-info

Il seguente esempio di codice mostra come utilizzare `get-connectivity-info`.

### AWS CLI

Per ottenere le informazioni sulla connettività per un dispositivo Greengrass core

L'`get-connectivity-info` esempio seguente ottiene le informazioni di connettività per un dispositivo core Greengrass. I dispositivi client utilizzano queste informazioni per connettersi al MQTT broker che funziona su questo dispositivo principale.

```
aws greengrassv2 get-connectivity-info \  
  --thing-name MyGreengrassCore
```

Output:

```
{  
  "connectivityInfo": [  
    {  
      "id": "localIP_192.0.2.0",  
      "hostAddress": "192.0.2.0",  
      "portNumber": 8883  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Manage core device endpoint](#) nella AWS IoT Greengrass V2 Developer Guide.

- Per API i dettagli, vedere [GetConnectivityInfo](#) in AWS CLI Command Reference.

## get-core-device

Il seguente esempio di codice mostra come utilizzare `get-core-device`.

## AWS CLI

Per ottenere un dispositivo principale

L'`get-core-device` seguente ottiene informazioni su un dispositivo principale AWS IoT Greengrass.

```
aws greengrassv2 get-core-device \  
  --core-device-thing-name MyGreengrassCore
```

Output:

```
{  
  "coreDeviceThingName": "MyGreengrassCore",  
  "coreVersion": "2.0.3",  
  "platform": "linux",  
  "architecture": "amd64",  
  "status": "HEALTHY",  
  "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00",  
  "tags": {}  
}
```

Per ulteriori informazioni, consulta [Verifica dello stato del dispositivo principale](#) nella Guida per AWS sviluppatori IoT Greengrass V2.

- Per API i dettagli, vedere [GetCoreDevice](#) in AWS CLI Command Reference.

## get-deployment

Il seguente esempio di codice mostra come utilizzare `get-deployment`.

## AWS CLI

Per ottenere una distribuzione

L'`get-deployment` seguente ottiene informazioni sull'implementazione del componente AWS IoT Greengrass nucleus su un gruppo di dispositivi principali.

```
aws greengrassv2 get-deployment \  
  --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

**Output:**

```
{
  "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": {}
}
```

Per ulteriori informazioni, consulta [Distribuire i componenti sui dispositivi](#) nella Guida per sviluppatori AWS IoT Greengrass V2.

- Per API i dettagli, consulta Command [GetDeployment](#)Reference AWS CLI .

## get-service-role-for-account

Il seguente esempio di codice mostra come utilizzare `get-service-role-for-account`.

### AWS CLI

Per ottenere il ruolo di servizio Greengrass per il tuo account AWS

L'`get-service-role-for-account`esempio seguente ottiene il ruolo di servizio associato a AWS IoT Greengrass per il tuo AWS account.

```
aws greengrassv2 get-service-role-for-account
```

Output:

```
{
  "associatedAt": "2022-01-19T19:21:53Z",
  "roleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
}
```

Per ulteriori informazioni, consulta il [ruolo del servizio Greengrass nella Guida](#) per sviluppatori IoT AWS Greengrass V2.

- Per API i dettagli, vedere [GetServiceRoleForAccount](#)in AWS CLI Command Reference.

## list-client-devices-associated-with-core-device

Il seguente esempio di codice mostra come utilizzare `list-client-devices-associated-with-core-device`.

### AWS CLI

Per elencare i dispositivi client associati a un dispositivo principale

L'`list-client-devices-associated-with-core-device`esempio seguente elenca tutti i dispositivi client associati a un dispositivo principale.

```
aws greengrassv2 list-client-devices-associated-with-core-device \
  --core-device-thing-name MyTestGreengrassCore
```



## Output:

```
{
  "associatedClientDevices": [
    {
      "thingName": "MyClientDevice2",
      "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
    },
    {
      "thingName": "MyClientDevice1",
      "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Interagire con i dispositivi IoT locali](#) nella Guida per AWS sviluppatori IoT Greengrass V2.

- Per API i dettagli, vedere [ListClientDevicesAssociatedWithCoreDevice](#) in AWS CLI Command Reference.

## list-component-versions

Il seguente esempio di codice mostra come utilizzare `list-component-versions`.

### AWS CLI

Per elencare le versioni di un componente

L'`list-component-versions` esempio seguente elenca tutte le versioni di un componente Hello World.

```
aws greengrassv2 list-component-versions \
  --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example>HelloWorld
```

## Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Manage components](#) nella AWS IoT Greengrass V2 Developer Guide.

- Per API i dettagli, vedere [ListComponentVersions](#) in AWS CLI Command Reference.

## list-components

Il seguente esempio di codice mostra come utilizzare `list-components`.

### AWS CLI

Per elencare i componenti

L'`list-components` esempio seguente elenca ogni componente e la sua versione più recente definita nell' AWS account nella regione corrente.

```
aws greengrassv2 list-components
```

Output:

```

{
  "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"
        }
      }
    ]
  }
]
```

Per ulteriori informazioni, consulta [Manage components](#) nella AWS IoT Greengrass V2 Developer Guide.

- Per API i dettagli, vedere [ListComponents](#) in AWS CLI Command Reference.

## list-core-devices

Il seguente esempio di codice mostra come utilizzare `list-core-devices`.

### AWS CLI

Per elencare i dispositivi principali

L'`list-core-devices` esempio seguente elenca i dispositivi principali AWS IoT Greengrass presenti nel tuo AWS account nella regione corrente.

```
aws greengrassv2 list-core-devices
```

Output:

```
{
  "coreDevices": [
    {
      "coreDeviceThingName": "MyGreengrassCore",
      "status": "HEALTHY",
      "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00"
    }
  ]
}
```

```
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Verifica dello stato del dispositivo principale](#) nella Guida per AWS sviluppatori IoT Greengrass V2.

- Per API i dettagli, vedere [ListCoreDevices](#) in AWS CLI Command Reference.

## list-deployments

Il seguente esempio di codice mostra come utilizzare `list-deployments`.

### AWS CLI

Per elencare le distribuzioni

L'`list-deployments` esempio seguente elenca la revisione più recente di ogni distribuzione definita nel tuo AWS account nella regione corrente.

```
aws greengrassv2 list-deployments
```

Output:

```
{  
  "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
    }
]
}
```

Per ulteriori informazioni, consulta [Distribuire i componenti sui dispositivi](#) nella Guida per sviluppatori AWS IoT Greengrass V2.

- Per API i dettagli, consulta Command [ListDeployments](#)Reference AWS CLI .

## list-effective-deployments

Il seguente esempio di codice mostra come utilizzare `list-effective-deployments`.

### AWS CLI

Per elencare i lavori di distribuzione

L'`list-effective-deployments` esempio seguente elenca le implementazioni che si applicano a un dispositivo core IoT AWS Greengrass.

```
aws greengrassv2 list-effective-deployments \
  --core-device-thing-name MyGreengrassCore
```

Output:

```
{
  "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"
  }
]
}
```

Per ulteriori informazioni, consulta [Verifica dello stato del dispositivo principale](#) nella Guida per AWS sviluppatori IoT Greengrass V2.

- Per API i dettagli, vedere [ListEffectiveDeployments](#) in AWS CLI Command Reference.

## list-installed-components

Il seguente esempio di codice mostra come utilizzare `list-installed-components`.

### AWS CLI

Per elencare i componenti installati su un dispositivo principale

L'`list-installed-components` esempio seguente elenca i componenti installati su un dispositivo core AWS IoT Greengrass.

```
aws greengrassv2 list-installed-components \
  --core-device-thing-name MyGreengrassCore
```

Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta [Verifica dello stato del dispositivo principale](#) nella Guida per AWS sviluppatori IoT Greengrass V2.

- Per API i dettagli, vedere [ListInstalledComponents](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una risorsa

L'`list-tags-for-resource` esempio seguente elenca tutti i tag per un dispositivo core AWS IoT Greengrass.

```
aws greengrassv2 list-tags-for-resource \
  --resource-arn arn:aws:greengrass:us-west-2:123456789012:coreDevices:MyGreengrassCore
```

Output:

```
{
  "tags": {
    "Owner": "richard-roe"
  }
}
```

Per ulteriori informazioni, consulta [Tagga le tue risorse](#) nella Guida per gli AWS sviluppatori di IoT Greengrass V2.

- Per API i dettagli, consulta AWS CLI Command [ListTagsForResource](#) Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere un tag a una risorsa

L'esempio seguente aggiunge un tag `owner` a un dispositivo core AWS IoT Greengrass. È possibile utilizzare questo tag per controllare l'accesso al dispositivo principale in base al proprietario.

```
aws greengrassv2 tag-resource \  
  --resource-arn arn:aws:greengrass:us-west-2:123456789012:coreDevices:MyGreengrassCore \  
  --tags Owner=richard-roe
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagga le tue risorse](#) nella Guida per gli AWS sviluppatori di IoT Greengrass V2.

- Per API i dettagli, consulta AWS CLI Command [TagResource](#) Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da una risorsa

L'esempio seguente rimuove un tag proprietario da un dispositivo principale AWS IoT Greengrass.

```
aws iotsitewise untag-resource \  
  --resource-arn arn:aws:greengrass:us-west-2:123456789012:coreDevices:MyGreengrassCore \  
  --tag-keys Owner
```

Questo comando non produce alcun output.



Per ulteriori informazioni, consulta [Tagga le tue risorse](#) nella Guida per gli AWS sviluppatori di IoT Greengrass V2.

- Per API i dettagli, consulta AWS CLI Command [UntagResource](#)Reference.

## update-connectivity-info

Il seguente esempio di codice mostra come utilizzare `update-connectivity-info`.

### AWS CLI

Per aggiornare le informazioni di connettività per un dispositivo Greengrass core

L'`update-connectivity-info`esempio seguente ottiene le informazioni di connettività per un dispositivo core Greengrass. I dispositivi client utilizzano queste informazioni per connettersi al MQTT broker che funziona su questo dispositivo principale.

```
aws greengrassv2 update-connectivity-info \  
  --thing-name MyGreengrassCore \  
  --cli-input-json file://core-device-connectivity-info.json
```

Contenuto di `core-device-connectivity-info.json`.

```
{  
  "connectivityInfo": [  
    {  
      "hostAddress": "192.0.2.0",  
      "portNumber": 8883,  
      "id": "localIP_192.0.2.0"  
    }  
  ]  
}
```

Output:

```
{  
  "version": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Per ulteriori informazioni, consulta [Manage core device endpoint](#) nella AWS IoT Greengrass V2 Developer Guide.

- Per API i dettagli, vedere [UpdateConnectivityInfo](#) in AWS CLI Command Reference.

## AWS IoT Jobs SDK release esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS IoT Jobs SDK release.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **describe-job-execution**

Il seguente esempio di codice mostra come utilizzare `describe-job-execution`.

AWS CLI

Per ottenere i dettagli dell'esecuzione di un lavoro

L'`describe-job-execution` esempio seguente recupera i dettagli dell'ultima esecuzione del processo e dell'oggetto specificati.

```
aws iot-jobs-data describe-job-execution \  
  --job-id SampleJob \  
  --thing-name MotionSensor1 \  
  --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

Output:

```
{  
  "execution": {
```

```
    "approximateSecondsBeforeTimedOut": 88,  
    "executionNumber": 2939653338,  
    "jobId": "SampleJob",  
    "lastUpdatedAt": 1567701875.743,  
    "queuedAt": 1567701902.444,  
    "status": "QUEUED",  
    "thingName": "MotionSensor1 ",  
    "versionNumber": 3  
  }  
}
```

Per ulteriori informazioni, consulta [Devices and Jobs](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [DescribeJobExecution](#) in AWS CLI Command Reference.

## get-pending-job-executions

Il seguente esempio di codice mostra come utilizzare `get-pending-job-executions`.

### AWS CLI

Per ottenere un elenco di tutti i lavori che non sono in stato di terminale per un oggetto

L'`get-pending-job-executions` seguente visualizza un elenco di tutti i lavori che non sono in uno stato terminale per l'oggetto specificato.

```
aws iot-jobs-data get-pending-job-executions \  
  --thing-name MotionSensor1  
  --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

Output:

```
{  
  "InProgressJobs": [  
  ],  
  "queuedJobs": [  
    {  
      "executionNumber": 2939653338,  
      "jobId": "SampleJob",  
      "lastUpdatedAt": 1567701875.743,  
      "queuedAt": 1567701902.444,  
      "versionNumber": 3  
    }  
  ]  
}
```

```
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Devices and Jobs](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [GetPendingJobExecutions](#) in AWS CLI Command Reference.

## start-next-pending-job-execution

Il seguente esempio di codice mostra come utilizzare `start-next-pending-job-execution`.

### AWS CLI

Per ottenere e avviare la successiva esecuzione del lavoro in sospeso per un oggetto

L'esempio seguente recupera e avvia l'esecuzione del lavoro successiva il cui stato è `IN_PROGRESS` o `QUEUED` per l'oggetto specificato.

```
aws iot-jobs-data start-next-pending-job-execution \  
  --thing-name MotionSensor1 \  
  --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

### Output:

```
{  
  "execution": {  
    "approximateSecondsBeforeTimedOut": 88,  
    "executionNumber": 2939653338,  
    "jobId": "SampleJob",  
    "lastUpdatedAt": 1567714853.743,  
    "queuedAt": 1567701902.444,  
    "startedAt": 1567714871.690,  
    "status": "IN_PROGRESS",  
    "thingName": "MotionSensor1 ",  
    "versionNumber": 3  
  }  
}
```

Per ulteriori informazioni, consulta [Devices and Jobs](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [StartNextPendingJobExecution](#) in AWS CLI Command Reference.

## update-job-execution

Il seguente esempio di codice mostra come utilizzare `update-job-execution`.

### AWS CLI

Per aggiornare lo stato dell'esecuzione di un lavoro

L'`update-job-execution`esempio seguente aggiorna lo stato del job e dell'oggetto specificati.

```
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
```

Output:

```
{  
  "executionState": {  
    "status": "REMOVED",  
    "versionNumber": 3  
  },  
}
```

Per ulteriori informazioni, consulta [Devices and Jobs](#) nella AWS IoT Developer Guide.

- Per API i dettagli, vedere [UpdateJobExecution](#) in AWS CLI Command Reference.

## AWS IoT SiteWise esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS IoT SiteWise.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **associate-assets**

Il seguente esempio di codice mostra come utilizzare `associate-assets`.

#### AWS CLI

Per associare una risorsa secondaria a una risorsa principale

L'esempio seguente associa una turbina eolica a una risorsa di un parco eolico, dove il modello degli asset delle turbine eoliche esiste come gerarchia nel modello degli asset del parco eolico.

```
aws iotsitewise associate-assets \  
  --asset-id a1b2c3d4-5678-90ab-cdef-4444EXAMPLE \  
  --hierarchy-id a1b2c3d4-5678-90ab-cdef-7777EXAMPLE \  
  --child-asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Associating assets](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [AssociateAssets AWS CLI Command Reference](#).

### **batch-associate-project-assets**

Il seguente esempio di codice mostra come utilizzare `batch-associate-project-assets`.

#### AWS CLI

Associare una risorsa a un progetto

L'esempio seguente associa un asset di un parco eolico a un progetto.

```
aws iotsitewise batch-associate-project-assets \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \  
  --asset-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE
```

```
--asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere risorse ai progetti](#) nella Guida all'applicazione AWS IoT SiteWise Monitor.

- Per API i dettagli, consulta [BatchAssociateProjectAssets AWS CLI Command Reference](#).

## **batch-disassociate-project-assets**

Il seguente esempio di codice mostra come utilizzare `batch-disassociate-project-assets`.

### AWS CLI

Dissociare una risorsa da un progetto

L'`batch-disassociate-project-assets` esempio seguente dissocia un asset di un parco eolico da un progetto.

```
aws iotsitewise batch-disassociate-project-assets \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \  
  --asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere risorse ai progetti](#) nella Guida all'applicazione AWS IoT SiteWise Monitor.

- Per API i dettagli, consulta [BatchDisassociateProjectAssets AWS CLI Command Reference](#).

## **batch-put-asset-property-value**

Il seguente esempio di codice mostra come utilizzare `batch-put-asset-property-value`.

### AWS CLI

Per inviare dati alle proprietà degli asset

L'`batch-put-asset-property-value` esempio seguente invia i dati di alimentazione e temperatura alle proprietà dell'asset identificate dagli alias di proprietà.

```
aws iotsitewise batch-put-asset-property-value \  
--cli-input-json file://batch-put-asset-property-value.json
```

Contenuto di `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  
          }  
        }  
      ]  
    }  
  ]  
}
```

Output:

```
{
```



```
"errorEntries": []
}
```

Per ulteriori informazioni, consulta [Ingestione di dati utilizzando l'IoT SiteWise API nella AWS IoT SiteWise User Guide](#).

- Per API i dettagli, vedere [BatchPutAssetPropertyValue](#) in AWS CLI Command Reference.

## create-access-policy

Il seguente esempio di codice mostra come utilizzare `create-access-policy`.

### AWS CLI

Esempio 1: concedere a un utente l'accesso amministrativo a un portale

L'esempio seguente crea una politica di accesso che concede a un utente l'accesso amministrativo a un portale Web per un'azienda di parchi eolici.

```
aws iotsitewise create-access-policy \
  --cli-input-json file://create-portal-administrator-access-policy.json
```

Contenuto di `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"
    }
  }
}
```

Output:

```
{
  "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE",
```

```
"accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE"
}
```

Per ulteriori informazioni, consulta [Aggiungere o rimuovere amministratori del portale](#) nella Guida per l' SiteWise utente AWS IoT.

Esempio 2: concedere a un utente l'accesso in sola lettura a un progetto

L'create-access-policyesempio seguente crea una politica di accesso che concede a un utente l'accesso in sola lettura a un progetto di parco eolico.

```
aws iotsitewise create-access-policy \
  --cli-input-json file://create-project-viewer-access-policy.json
```

Contenuto di create-project-viewer-access-policy.json.

```
{
  "accessPolicyIdentity": {
    "user": {
      "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
    }
  },
  "accessPolicyPermission": "VIEWER",
  "accessPolicyResource": {
    "project": {
      "id": "a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE"
    }
  }
}
```

Output:

```
{
  "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE",
  "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE"
}
```

Per ulteriori informazioni, consulta [Assegnazione dei visualizzatori di progetti nella Guida](#) all'applicazione AWS IoT SiteWise Monitor.

- Per API i dettagli, consulta AWS CLI Command [CreateAccessPolicyReference](#).

## create-asset-model

Il seguente esempio di codice mostra come utilizzare `create-asset-model`.

### AWS CLI

Per creare un modello di asset

L'`create-asset-model` esempio seguente crea un modello di asset che definisce una turbina eolica con le seguenti proprietà:

Numero di serie - Il numero di serie di un' turbina  
Generated energia eolica - Il flusso di dati sull'energia generata da un vento  
turbineTemperature C - Il flusso di dati sulla temperatura di una turbina eolica in Celsius  
Temperature F - I dati di temperatura mappati vanno da Celsius a Fahrenheit

```
aws iotsitewise create-asset-model \  
  --cli-input-json file://create-wind-turbine-model.json
```

Contenuto di `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"
      }
    }
  }
}
```

```

    }
  }
}
]
}

```

Output:

```

{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelStatus": {
    "state": "CREATING"
  }
}

```

Per ulteriori informazioni, consulta [Definizione dei modelli di asset](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, vedere [CreateAssetModel](#) in AWS CLI Command Reference.

## create-asset

Il seguente esempio di codice mostra come utilizzare `create-asset`.

### AWS CLI

Per creare una risorsa

L'`create-asset` esempio seguente crea un asset di turbine eoliche a partire da un modello di asset di turbine eoliche.

```

aws iotsitewise create-asset \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
  --asset-name "Wind Turbine 1"

```

Output:

```

{

```

```
"assetId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
"assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
"assetStatus": {
  "state": "CREATING"
}
}
```

Per ulteriori informazioni, consulta [Creazione di risorse](#) nella Guida per SiteWise l'utente AWS IoT.

- Per API i dettagli, consulta [CreateAsset AWS CLI](#) Command Reference.

## create-dashboard

Il seguente esempio di codice mostra come utilizzare `create-dashboard`.

### AWS CLI

Per creare una dashboard

L'`create-dashboard` esempio seguente crea una dashboard con un grafico a linee che mostra la potenza totale generata per un parco eolico.

```
aws iotsitewise create-dashboard \
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \
  --dashboard-name "Wind Farm" \
  --dashboard-definition file://create-wind-farm-dashboard.json
```

Contenuto di `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"
      }
    ]
  }
}

```

Output:

```

{
  "dashboardId": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
  "dashboardArn": "arn:aws:iotsitewise:us-west-2:123456789012:dashboard/a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE"
}

```

Per ulteriori informazioni, consulta [Creating dashboards \(CLI\)](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [CreateDashboard AWS CLI Command Reference](#).

## create-gateway

Il seguente esempio di codice mostra come utilizzare `create-gateway`.

### AWS CLI

Per creare un gateway

L'esempio seguente crea un gateway che funziona su AWS IoT Greengrass.

```

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}

```

Output:

```

{

```

```
"gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
"gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
}
```

Per ulteriori informazioni, consulta [Configurazione di un gateway](#) nella Guida per l'utente AWS IoT SiteWise.

- Per API i dettagli, vedere [CreateGateway](#) in AWS CLI Command Reference.

## create-portal

Il seguente esempio di codice mostra come utilizzare `create-portal`.

### AWS CLI

Per creare un portale

L'esempio seguente crea un portale web per un'azienda di parchi eolici. È possibile creare portali solo nella stessa regione in cui è stato abilitato il AWS Single Sign-On.

```
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
```

Output:

```
{
  "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"
}
```



Per ulteriori informazioni, consulta la sezione Guida [introduttiva a AWS IoT SiteWise Monitor](#) nella AWS IoT SiteWise User Guide e [Enabling AWS SSO](#) in the AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [CreatePortal AWS CLI](#) Command Reference.

## create-project

Il seguente esempio di codice mostra come utilizzare `create-project`.

### AWS CLI

Per creare un progetto

L'`create-project` esempio seguente crea un progetto di parco eolico.

```
aws iotsitewise create-project \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE \  
  --project-name "Wind Farm 1" \  
  --project-description "Contains asset visualizations for Wind Farm #1 for Example Corp."
```

Output:

```
{  
  "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE",  
  "projectArn": "arn:aws:iotsitewise:us-west-2:123456789012:project/  
a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE"  
}
```

Per ulteriori informazioni, consulta [Creazione di progetti](#) nella Guida all'applicazione AWS IoT SiteWise Monitor.

- Per API i dettagli, consulta [CreateProject AWS CLI](#) Command Reference.

## delete-access-policy

Il seguente esempio di codice mostra come utilizzare `delete-access-policy`.

### AWS CLI

Per revocare l'accesso di un utente a un progetto o portale

L'`delete-access-policy` esempio seguente elimina una politica di accesso che concede a un utente l'accesso amministrativo a un portale.

```
aws iotsitewise delete-access-policy \  
  --access-policy-id a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere o rimuovere amministratori del portale](#) nella Guida per l' SiteWise utente AWS IoT.

- Per API i dettagli, vedere [DeleteAccessPolicy](#) in AWS CLI Command Reference.

## `delete-asset-model`

Il seguente esempio di codice mostra come utilizzare `delete-asset-model`.

### AWS CLI

Per eliminare un modello di asset

L'`delete-asset-model` esempio seguente elimina un modello di asset per turbine eoliche.

```
aws iotsitewise delete-asset-model \  
  --asset-model-id a1b2c3d4-5678-90ab-cdef-1111EXAMPLE
```

Output:

```
{  
  "assetModelStatus": {  
    "state": "DELETING"  
  }  
}
```

Per ulteriori informazioni, consulta [Eliminazione dei modelli di asset](#) nella Guida per l' SiteWise utente AWS IoT.

- Per API i dettagli, consulta [DeleteAssetModel AWS CLI](#) Command Reference.

## `delete-asset`

Il seguente esempio di codice mostra come utilizzare `delete-asset`.

## AWS CLI

Per eliminare una risorsa

L'`delete-asset` seguente elimina un asset di una turbina eolica.

```
aws iotsitewise delete-asset \  
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE
```

Output:

```
{  
  "assetStatus": {  
    "state": "DELETING"  
  }  
}
```

Per ulteriori informazioni, consulta [Eliminazione delle risorse](#) nella Guida per l'utente SiteWise di AWS IoT.

- Per API e i dettagli, consulta [DeleteAsset AWS CLI Command Reference](#).

## `delete-dashboard`

Il seguente esempio di codice mostra come utilizzare `delete-dashboard`.

### AWS CLI

Per eliminare un pannello di controllo

L'`delete-dashboard` seguente elimina il cruscotto di una turbina eolica.

```
aws iotsitewise delete-dashboard \  
  --dashboard-id a1b2c3d4-5678-90ab-cdef-ffffEXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione dei dashboard nella Guida](#) all'applicazione AWS IoT SiteWise Monitor.

- Per API e i dettagli, consulta [AWS CLI Command DeleteDashboard Reference](#).

## delete-gateway

Il seguente esempio di codice mostra come utilizzare `delete-gateway`.

### AWS CLI

Per eliminare un gateway

L'`delete-gateway` esempio seguente elimina un gateway.

```
aws iotsitewise delete-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Ingestione di dati utilizzando un gateway](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, vedere [DeleteGateway](#) in AWS CLI Command Reference.

## delete-portal

Il seguente esempio di codice mostra come utilizzare `delete-portal`.

### AWS CLI

Per eliminare un portale

L'`delete-portal` esempio seguente elimina un portale Web per una società di parchi eolici.

```
aws iotsitewise delete-portal \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

Output:

```
{  
  "portalStatus": {  
    "state": "DELETING"  
  }  
}
```

Per ulteriori informazioni, consulta [Eliminazione di un portale](#) nella Guida per l'utente SiteWise AWS IoT.

- Per API i dettagli, vedere [DeletePortal](#) in AWS CLI Command Reference.

## delete-project

Il seguente esempio di codice mostra come utilizzare `delete-project`.

### AWS CLI

Per eliminare un progetto

L'`delete-project` esempio seguente elimina un progetto di parco eolico.

```
aws iotsitewise delete-project \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di progetti](#) nella Guida all'applicazione AWS IoT SiteWise Monitor.

- Per API i dettagli, consulta [DeleteProject AWS CLI](#) Command Reference.

## describe-access-policy

Il seguente esempio di codice mostra come utilizzare `describe-access-policy`.

### AWS CLI

Per descrivere una politica di accesso

L'`describe-access-policy` esempio seguente descrive una politica di accesso che concede a un utente l'accesso amministrativo a un portale Web per un'azienda di parchi eolici.

```
aws iotsitewise describe-access-policy \  
  --access-policy-id a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE
```

Output:

```
{  
  "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-bbbbbbEXAMPLE"
  }
},
"accessPolicyResource": {
  "portal": {
    "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"
  }
},
"accessPolicyPermission": "ADMINISTRATOR",
"accessPolicyCreationDate": "2020-02-20T22:35:15.552880124Z",
"accessPolicyLastUpdateDate": "2020-02-20T22:35:15.552880124Z"
}

```

Per ulteriori informazioni, consulta [Aggiungere o rimuovere amministratori del portale](#) nella Guida per l' SiteWise utente AWS IoT.

- Per API i dettagli, vedere [DescribeAccessPolicy](#) in AWS CLI Command Reference.

## describe-asset-model

Il seguente esempio di codice mostra come utilizzare `describe-asset-model`.

### AWS CLI

Per descrivere un modello di asset

L'`describe-asset-model` esempio seguente descrive un modello di asset per un parco eolico.

```

aws iotsitewise describe-asset-model \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE

```

Output:

```

{
  "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"
    }
  }
}

```

Per ulteriori informazioni, consulta [Descrivere un modello di asset specifico](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [DescribeAssetModel AWS CLI Command Reference](#).

## describe-asset-property

Il seguente esempio di codice mostra come utilizzare `describe-asset-property`.

### AWS CLI

Per descrivere la proprietà di un asset

L'esempio seguente descrive la proprietà della potenza totale generata di un impianto eolico.

```

aws iotsitewise describe-asset-property \
  --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
  --property-id a1b2c3d4-5678-90ab-cdef-99999EXAMPLE

```

Output:

```

{
  "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"
    }
  }
}

```





**Output:**

```
{
  "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
  "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
  "assetName": "Wind Farm 1",
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetProperties": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
      "name": "Region",
      "dataType": "STRING"
    },
    {
      "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"
  }
}
```

Per ulteriori informazioni, consulta [Descrivere un asset specifico](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [DescribeAsset AWS CLI Command Reference](#).

**describe-dashboard**

Il seguente esempio di codice mostra come utilizzare `describe-dashboard`.

## AWS CLI

Per descrivere una dashboard

L'`describe-dashboard`esempio seguente descrive il pannello di controllo del parco eolico specificato.

```
aws iotsitewise describe-dashboard \
  --dashboard-id a1b2c3d4-5678-90ab-cdef-ffffEXAMPLE
```

Output:

```
{
  "dashboardId": "a1b2c3d4-5678-90ab-cdef-ffffEXAMPLE",
  "dashboardArn": "arn:aws:iotsitewise:us-west-2:123456789012:dashboard/a1b2c3d4-5678-90ab-cdef-ffffEXAMPLE",
  "dashboardName": "Wind Farm",
  "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE",
  "dashboardDefinition": "{\n  \"widgets\": [\n    {\n      \"type\": \"monitor-line-chart\",\n      \"title\": \"Generated Power\",\n      \"x\": 0,\n      \"y\": 0,\n      \"height\": 3,\n      \"width\": 3,\n      \"metrics\": [\n        {\n          \"label\": \"Power\",\n          \"type\": \"iotsitewise\",\n          \"assetId\": \"a1b2c3d4-5678-90ab-cdef-4444EXAMPLE\",\n          \"propertyId\": \"a1b2c3d4-5678-90ab-cdef-9999EXAMPLE\"\n        }\n      ]\n    }\n  ]\n}",
  "dashboardCreationDate": "2020-05-01T20:32:12.228476348Z",
  "dashboardLastUpdateDate": "2020-05-01T20:32:12.228476348Z"
}
```

Per ulteriori informazioni, consulta [Visualizzazione dei dashboard](#) nella Guida all'applicazione AWS IoT SiteWise Monitor.

- Per API i dettagli, consulta [DescribeDashboard AWS CLI Command Reference](#).

## `describe-gateway-capability-configuration`

Il seguente esempio di codice mostra come utilizzare `describe-gateway-capability-configuration`.

### AWS CLI

Per descrivere una funzionalità gateway

L'`describe-gateway-capability-configuration`esempio seguente descrive una funzionalità di origine OPC-UA.

```
aws iotsitewise describe-gateway-capability-configuration \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
  --capability-namespace "iotsitewise:opcuacollector:1"
```

Output:

```
{
  "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:greengrass-factory1-auth-3QNDmM\\"},\\"nodeFilterRules\\":
[]},\\"measurementDataStreamPrefix\\":\\"\\"}]}",
  "capabilitySyncStatus": "IN_SYNC"
}
```

Per ulteriori informazioni, consulta [Configurazione delle fonti di dati](#) nella Guida per l'utente AWS IoT.

- Per API i dettagli, vedere [DescribeGatewayCapabilityConfiguration](#) in AWS CLI Command Reference.

## describe-gateway

Il seguente esempio di codice mostra come utilizzare `describe-gateway`.

AWS CLI

Per descrivere un gateway

L'esempio seguente descrive un gateway.

```
aws iotsitewise describe-gateway \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Output:

```
{
```

```
"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
}
```

Per ulteriori informazioni, consulta [Ingestione di dati utilizzando un gateway](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, vedere [DescribeGateway](#) in AWS CLI Command Reference.

## describe-logging-options

Il seguente esempio di codice mostra come utilizzare `describe-logging-options`.

AWS CLI

Per recuperare le opzioni di SiteWise registrazione AWS IoT correnti

L'`describe-logging-options`esempio seguente recupera le opzioni di SiteWise registrazione AWS IoT correnti per il tuo AWS account nella regione corrente.

```
aws iotsitewise describe-logging-options
```

Output:

```
{
```

```
"loggingOptions": {
  "level": "INFO"
}
}
```

Per ulteriori informazioni, consulta [Monitoring AWS IoT SiteWise with Amazon CloudWatch Logs](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [DescribeLoggingOptions AWS CLI Command Reference](#).

## describe-portal

Il seguente esempio di codice mostra come utilizzare `describe-portal`.

### AWS CLI

Per descrivere un portale

L'`describe-portal` esempio seguente descrive un portale web per un'azienda di parchi eolici.

```
aws iotsitewise describe-portal \
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

Output:

```
{
  "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"
```

```
}
```

Per ulteriori informazioni, consulta [Amministrare i portali nella Guida](#) per SiteWise l'utente AWS IoT.

- Per API i dettagli, vedere [DescribePortal](#) in AWS CLI Command Reference.

## describe-project

Il seguente esempio di codice mostra come utilizzare `describe-project`.

### AWS CLI

Per descrivere un progetto

L'esempio seguente descrive un progetto di parco eolico.

```
aws iotsitewise describe-project \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE
```

Output:

```
{  
  "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE",  
  "projectArn": "arn:aws:iotsitewise:us-west-2:123456789012:project/  
a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE",  
  "projectName": "Wind Farm 1",  
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaEXAMPLE",  
  "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"  
}
```

Per ulteriori informazioni, consulta [Visualizzazione dei dettagli del progetto](#) nella Guida all'applicazione AWS IoT SiteWise Monitor.

- Per API i dettagli, vedere [DescribeProject](#) in AWS CLI Command Reference.

## disassociate-assets

Il seguente esempio di codice mostra come utilizzare `disassociate-assets`.

## AWS CLI

Per dissociare una risorsa figlia da una risorsa principale

L'`disassociate-assets` seguente dissocia una turbina eolica da una risorsa di un parco eolico.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Associating assets](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [DisassociateAssets AWS CLI Command Reference](#).

## get-asset-property-aggregates

Il seguente esempio di codice mostra come utilizzare `get-asset-property-aggregates`.

### AWS CLI

Per recuperare la media aggregata e i valori di conteggio di una proprietà di asset

L'`get-asset-property-aggregates` seguente recupera la potenza totale media di una turbina eolica e il conteggio dei punti dati sulla potenza totale per un periodo di 1 ora.

```
aws iotsitewise get-asset-property-aggregates \  
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \  
  --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE \  
  --start-date 1580849400 \  
  --end-date 1580853000 \  
  --aggregate-types AVERAGE COUNT \  
  --resolution 1h
```

Output:

```
{  
  "aggregatedValues": [  
    {  
      "timestamp": 1580850000.0,
```



```

        "quality": "GOOD",
        "value": {
            "average": 8723.46538886233,
            "count": 12.0
        }
    }
]
}

```

Per ulteriori informazioni, consulta [Interrogazione degli aggregati di proprietà degli asset](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetAssetPropertyAggregates](#)Reference.

## get-asset-property-value-history

Il seguente esempio di codice mostra come utilizzare `get-asset-property-value-history`.

### AWS CLI

Per recuperare i valori storici di una proprietà di asset

L'`get-asset-property-value-history`esempio seguente recupera i valori di potenza totale di una turbina eolica per un periodo di 20 minuti.

```

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

```

Output:

```

{
  "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"
  }
]
}
```

Per ulteriori informazioni, consulta [Interrogazione dei valori storici delle proprietà degli asset](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [GetAssetPropertyValueHistory AWS CLI Command Reference](#).

## get-asset-property-value

Il seguente esempio di codice mostra come utilizzare `get-asset-property-value`.

## AWS CLI

Per recuperare il valore corrente di una proprietà dell'asset

L'`get-asset-property-value` seguente recupera la potenza totale corrente di una turbina eolica.

```
aws iotsitewise get-asset-property-value \  
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \  
  --property-id a1b2c3d4-5678-90ab-cdef-6666EXAMPLE
```

Output:

```
{  
  "propertyValue": {  
    "value": {  
      "doubleValue": 6890.8677520453875  
    },  
    "timestamp": {  
      "timeInSeconds": 1580853000,  
      "offsetInNanos": 0  
    },  
    "quality": "GOOD"  
  }  
}
```

Per ulteriori informazioni, consulta [Interrogazione dei valori delle proprietà degli asset correnti](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [GetAssetPropertyValue AWS CLI](#) Command Reference.

## list-access-policies

Il seguente esempio di codice mostra come utilizzare `list-access-policies`.

### AWS CLI

Per elencare tutte le politiche di accesso

L'`list-access-policies` seguente elenca tutte le politiche di accesso per un utente che è amministratore del portale.

```
aws iotsitewise list-access-policies \  
  --identity-type USER \  
  --identity-id a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE
```

Output:

```
{  
  "accessPolicySummaries": [  
    {  
      "id": "a1b2c3d4-5678-90ab-cdef-ccccEXAMPLE",  
      "identity": {  
        "user": {  
          "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"  
        }  
      },  
      "resource": {  
        "portal": {  
          "id": "a1b2c3d4-5678-90ab-cdef-aaaaEXAMPLE"  
        }  
      },  
      "permission": "ADMINISTRATOR"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Amministrare i portali nella Guida](#) per SiteWise l'utente AWS IoT.

- Per API i dettagli, vedere [ListAccessPolicies](#) in AWS CLI Command Reference.

## list-asset-models

Il seguente esempio di codice mostra come utilizzare `list-asset-models`.

AWS CLI

Per elencare tutti i modelli di asset

L'`list-asset-models` esempio seguente elenca tutti i modelli di asset definiti nel vostro AWS account nella regione corrente.

```
aws iotsitewise list-asset-models
```

**Output:**

```
{
  "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"
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta [Elenco di tutti i modelli di asset](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [ListAssetModels AWS CLI Command Reference](#).

**list-assets**

Il seguente esempio di codice mostra come utilizzare `list-assets`.

## AWS CLI

Esempio 1: per elencare tutte le risorse di primo livello

L'`list-assets` seguente elenca tutti gli asset che sono di primo livello nell'albero della gerarchia degli asset e definiti nel tuo AWS account nella regione corrente.

```
aws iotsitewise list-assets \  
  --filter TOP_LEVEL
```

Output:

```
{  
  "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"  
        }  
      ]  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Elencare gli asset](#) nella AWS IoT SiteWise User Guide.

Esempio 2: per elencare tutti gli asset in base a un modello di asset

L'`list-assets` seguente elenca tutti gli asset basati su un modello di asset e definiti nel tuo AWS account nella regione corrente.

```
aws iotsitewise list-assets \  
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Output:

```
{  
  "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": []  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Elencare gli asset](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [ListAssets AWS CLI](#) Command Reference.

## list-associated-assets

Il seguente esempio di codice mostra come utilizzare `list-associated-assets`.

### AWS CLI

Per elencare tutte le risorse associate a una risorsa in una gerarchia specifica

L'`list-associated-assets` esempio seguente elenca tutti gli asset delle turbine eoliche associati all'asset del parco eolico specificato.

```
aws iotsitewise list-associated-assets \  
  --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \  
  --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE
```

**Output:**

```
{
  "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": []
    }
  ]
}
```

Per ulteriori informazioni, consulta [Elencare gli asset associati a un asset specifico](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [ListAssociatedAssets AWS CLI](#) Command Reference.

**list-dashboards**

Il seguente esempio di codice mostra come utilizzare `list-dashboards`.

**AWS CLI**

Per elencare tutte le dashboard di un progetto

L'`list-dashboards` esempio seguente elenca tutte le dashboard definite in un progetto.

```
aws iotsitewise list-dashboards \
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE
```

**Output:**

```
{
```



```
"dashboardSummaries": [  
  {  
    "id": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",  
    "name": "Wind Farm",  
    "creationDate": "2020-05-01T20:32:12.228476348Z",  
    "lastUpdateDate": "2020-05-01T20:32:12.228476348Z"  
  }  
]  
}
```

Per ulteriori informazioni, consulta [Visualizzazione dei dashboard](#) nella Guida all'applicazione AWS IoT SiteWise Monitor.

- Per API i dettagli, consulta [ListDashboards AWS CLI](#) Command Reference.

## list-gateways

Il seguente esempio di codice mostra come utilizzare `list-gateways`.

### AWS CLI

Per elencare tutti i gateway

L'`list-gateways` esempio seguente elenca tutti i gateway definiti nell' AWS account nella regione corrente.

```
aws iotsitewise list-gateways
```

Output:

```
{  
  "gatewaySummaries": [  
    {  
      "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",  
      "gatewayName": "ExampleCorpGateway",  
      "gatewayCapabilitySummaries": [  
        {  
          "capabilityNamespace": "iotsitewise:opcuacollector:1",  
          "capabilitySyncStatus": "IN_SYNC"  
        }  
      ],  
    }  
  ],  
}
```

```
        "creationDate": 1588369971.457,  
        "lastUpdateDate": 1588369971.457  
    }  
]  
}
```

Per ulteriori informazioni, consulta [Ingestione di dati utilizzando un gateway](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, vedere [ListGateways](#) in AWS CLI Command Reference.

## list-portals

Il seguente esempio di codice mostra come utilizzare `list-portals`.

### AWS CLI

Per elencare tutti i portali

L'`list-portals` esempio seguente elenca tutti i portali definiti nel tuo AWS account nella regione corrente.

```
aws iotsitewise list-portals
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

```
}
```

Per ulteriori informazioni, consulta [Amministrare i portali nella Guida](#) per SiteWise l'utente AWS IoT.

- Per API i dettagli, vedere [ListPortals](#) in AWS CLI Command Reference.

## list-project-assets

Il seguente esempio di codice mostra come utilizzare `list-project-assets`.

### AWS CLI

Per elencare tutte le risorse associate a un progetto

L'`list-project-assets` esempio seguente elenca tutti gli asset associati a un progetto di parco eolico.

```
aws iotsitewise list-projects \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE
```

Output:

```
{  
  "assetIds": [  
    "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"  
  ]  
}
```

Per ulteriori informazioni, consulta [Aggiungere risorse ai progetti](#) nella Guida all'applicazione AWS IoT SiteWise Monitor.

- Per API i dettagli, consulta [ListProjectAssets AWS CLI](#) Command Reference.

## list-projects

Il seguente esempio di codice mostra come utilizzare `list-projects`.

### AWS CLI

Per elencare tutti i progetti in un portale

L'`list-projects` seguente elenca tutti i progetti definiti in un portale.

```
aws iotsitewise list-projects \  
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

Output:

```
{  
  "projectSummaries": [  
    {  
      "id": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE",  
      "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Visualizzazione dei dettagli del progetto](#) nella Guida all'applicazione AWS IoT SiteWise Monitor.

- Per API i dettagli, vedere [ListProjects](#) in AWS CLI Command Reference.

## `list-tags-for-resource`

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

AWS CLI

Per elencare tutti i tag di una risorsa

L'`list-tags-for-resource` seguente elenca tutti i tag relativi a una turbina eolica.

```
aws iotsitewise list-tags-for-resource \  
  --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

Output:

```
{
  "tags": {
    "Owner": "richard-roe"
  }
}
```

Per ulteriori informazioni, consulta [Tagging your resources](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## put-logging-options

Il seguente esempio di codice mostra come utilizzare `put-logging-options`.

### AWS CLI

Per specificare il livello di registrazione

L'`put-logging-options` esempio seguente abilita la registrazione dei INFO livelli in AWS IoT SiteWise. Gli altri livelli includono DEBUG e OFF

```
aws iotsitewise put-logging-options \
  --logging-options Level=INFO
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Monitoring AWS IoT SiteWise with Amazon CloudWatch Logs](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [PutLoggingOptions AWS CLI Command Reference](#).

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere un tag a una risorsa

L'`tag-resource` esempio seguente aggiunge un tag proprietario a una turbina eolica. Ciò consente di controllare l'accesso alla risorsa in base a chi la possiede.

```
aws iotsitewise tag-resource \  
  --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \  
  --tags Owner=richard-roe
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging your resources](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [TagResource AWS CLI](#) Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da una risorsa

L'esempio seguente rimuove un tag proprietario da una risorsa di una turbina eolica.

```
aws iotsitewise untag-resource \  
  --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/  
a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \  
  --tag-keys Owner
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging your resources](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [UntagResource AWS CLI](#) Command Reference.

## update-access-policy

Il seguente esempio di codice mostra come utilizzare `update-access-policy`.

### AWS CLI

Per concedere a un visualizzatore del progetto la proprietà di un progetto

L'update-access-policy esempio seguente aggiorna una politica di accesso che concede a un visualizzatore del progetto la proprietà di un progetto.

```
aws iotsitewise update-access-policy \  
  --access-policy-id a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE \  
  --cli-input-json file://update-project-viewer-access-policy.json
```

Contenuto di update-project-viewer-access-policy.json.

```
{  
  "accessPolicyIdentity": {  
    "user": {  
      "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbbEXAMPLE"  
    }  
  },  
  "accessPolicyPermission": "ADMINISTRATOR",  
  "accessPolicyResource": {  
    "project": {  
      "id": "a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE"  
    }  
  }  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Assegnazione dei proprietari dei progetti](#) nella Guida all'applicazione AWS IoT SiteWise Monitor.

- Per API i dettagli, consulta [UpdateAccessPolicy AWS CLI Command Reference](#).

## update-asset-model

Il seguente esempio di codice mostra come utilizzare `update-asset-model`.

### AWS CLI

Per aggiornare un modello di asset

L'update-asset-model esempio seguente aggiorna la descrizione del modello di impianto di un parco eolico. Questo esempio include il modello esistente IDs e le definizioni, poiché `update-asset-model` sovrascrive il modello esistente con il nuovo modello.

```
aws iotsitewise update-asset-model \  
--cli-input-json file://update-wind-farm-model.json
```

Contenuto di `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"
    }
  ]
}

```

Output:

```

{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetModelStatus": {
    "state": "CREATING"
  }
}

```

Per ulteriori informazioni, consulta [Updating asset models](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [UpdateAssetModel AWS CLI Command Reference](#).

## update-asset-property

Il seguente esempio di codice mostra come utilizzare `update-asset-property`.

### AWS CLI

Esempio 1: per aggiornare l'alias di una proprietà di asset

L'`update-asset-property` esempio seguente aggiorna l'alias della proprietà di alimentazione di un asset di una turbina eolica.

```

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

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Mappatura dei flussi di dati industriali alle proprietà degli asset](#) nella AWS IoT SiteWise User Guide.

Esempio 2: per abilitare le notifiche sulle proprietà degli asset

L'update-asset-propertyesempio seguente abilita le notifiche di aggiornamento delle proprietà energetiche di una turbina eolica. Gli aggiornamenti dei valori delle proprietà vengono pubblicati nell'MQTTargomento\$aws/sitewise/asset-models/<assetModelId>/assets/<assetId>/properties/<propertyId>, dove ogni ID viene sostituito dalla proprietà, dalla risorsa e dall'ID del modello della proprietà dell'asset.

```
aws iotsitewise update-asset-property \  
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \  
  --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE \  
  --property-notification-state ENABLED \  
  --property-alias "/examplecorp/windfarm/1/turbine/1/power"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Interagire con altri servizi](#) nella Guida per l' SiteWise utente AWS IoT.

- Per API i dettagli, vedere [UpdateAssetProperty](#) in AWS CLI Command Reference.

## update-asset

Il seguente esempio di codice mostra come utilizzareupdate-asset.

### AWS CLI

Per aggiornare il nome di una risorsa

L'update-assetesempio seguente aggiorna il nome di una turbina eolica.

```
aws iotsitewise update-asset \  
  --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \  
  --asset-name "Wind Turbine 2"
```

Output:

```
{
  "assetStatus": {
    "state": "UPDATING"
  }
}
```

Per ulteriori informazioni, consulta [Updating assets](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [UpdateAsset AWS CLI](#) Command Reference.

## update-dashboard

Il seguente esempio di codice mostra come utilizzare `update-dashboard`.

### AWS CLI

Per aggiornare un pannello di controllo

L'`update-dashboard` esempio seguente modifica il titolo del grafico a linee di un pannello di controllo che mostra la potenza totale generata per un parco eolico.

```
aws iotsitewise update-dashboard \
  --project-id a1b2c3d4-5678-90ab-cdef-ffffEXAMPLE \
  --dashboard-name "Wind Farm" \
  --dashboard-definition file://update-wind-farm-dashboard.json
```

Contenuto di `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"
  }
]
}

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creating dashboards \(CLI\)](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, consulta [UpdateDashboard AWS CLI Command Reference](#).

## update-gateway-capability-configuration

Il seguente esempio di codice mostra come utilizzare `update-gateway-capability-configuration`.

### AWS CLI

Per aggiornare la funzionalità di un gateway

L'`update-gateway-capability-configuration` esempio seguente configura una sorgente OPC-UA con le seguenti proprietà:

Considera attendibile qualsiasi certificato. Utilizza l'algoritmo Basic256 per proteggere i messaggi. Utilizza la SignAndEncrypt modalità per proteggere le connessioni. Utilizza le credenziali di autenticazione memorizzate in un segreto di Secrets Manager. 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

```

Contenuto di `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:greengrass-windfarm1-auth-1ABCDE"
      },
      "nodeFilterRules": []
    },
    "measurementDataStreamPrefix": ""
  }
]
}

```

Output:

```

{
  "capabilityNamespace": "iotsitewise:opcuacollector:1",
  "capabilitySyncStatus": "OUT_OF_SYNC"
}

```

Per ulteriori informazioni, consulta [Configurazione delle fonti di dati](#) nella Guida per l'utente AWS IoT SiteWise.

- Per API i dettagli, vedere [UpdateGatewayCapabilityConfiguration](#) in AWS CLI Command Reference.

## update-gateway

Il seguente esempio di codice mostra come utilizzare `update-gateway`.

### AWS CLI

Per aggiornare il nome di un gateway

L'esempio seguente aggiorna il nome di un gateway.

```
aws iotsitewise update-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \  
  --gateway-name ExampleCorpGateway1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Ingestione di dati utilizzando un gateway](#) nella AWS IoT SiteWise User Guide.

- Per API i dettagli, vedere [UpdateGateway](#) in AWS CLI Command Reference.

## update-portal

Il seguente esempio di codice mostra come utilizzare `update-portal`.

### AWS CLI

Per aggiornare i dettagli di un portale

L'`update-portal` esempio seguente aggiorna un portale web per un'azienda di parchi eolici.

```
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
```

Output:

```
{  
  "portalStatus": {  
    "state": "UPDATING"  
  }  
}
```

Per ulteriori informazioni, consulta [Amministrare i portali nella Guida](#) per SiteWise l'utente AWS IoT.

- Per API i dettagli, vedere [UpdatePortal](#) in AWS CLI Command Reference.

## update-project

Il seguente esempio di codice mostra come utilizzare `update-project`.

### AWS CLI

Per aggiornare i dettagli di un progetto

L'`update-project` esempio seguente aggiorna un progetto di parco eolico.

```
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."
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Modifica dei dettagli del progetto](#) nella Guida all'applicazione AWS IoT SiteWise Monitor.

- Per API i dettagli, consulta [UpdateProject AWS CLI](#) Command Reference.

## AWS IoT Things Graph esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS IoT Things Graph.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **associate-entity-to-thing**

Il seguente esempio di codice mostra come utilizzare `associate-entity-to-thing`.

#### AWS CLI

Associare un oggetto a un dispositivo

L'`associate-entity-to-thing`esempio seguente associa un oggetto a un dispositivo.

L'esempio utilizza un dispositivo con sensore di movimento che si trova nel namespace pubblico.

```
aws iotthingsgraph associate-entity-to-thing \  
  --thing-name "MotionSensorName" \  
  --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creazione e caricamento di modelli](#) nella Guida per l'utente di AWS IoT Things Graph.

- Per API i dettagli, vedere [AssociateEntityToThing](#) in AWS CLI Command Reference.

### **create-flow-template**

Il seguente esempio di codice mostra come utilizzare `create-flow-template`.

#### AWS CLI

Per creare un flusso

L'`create-flow-template`esempio seguente crea un flusso (workflow). Il valore di `MyFlowDefinition` è il GraphQL che modella il flusso.

```
aws iotthingsgraph create-flow-template \  
  --definition language=GRAPHQL,text="MyFlowDefinition"
```

Output:

```
{  
  "summary": {
```



```

    "createdAt": 1559248067.545,
    "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
    "revisionNumber": 1
  }
}

```

Per ulteriori informazioni, consulta [Working with Flows](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [CreateFlowTemplate](#) in AWS CLI Command Reference.

## create-system-instance

Il seguente esempio di codice mostra come utilizzare `create-system-instance`.

### AWS CLI

Per creare un'istanza di sistema

L'esempio seguente crea un'istanza di sistema. Il valore di `MySystemInstanceDefinition` è il GraphQL che modella l'istanza di sistema.

```

aws iotthingsgraph create-system-instance -\
  -definition language=GRAPHQL,text="MySystemInstanceDefinition" \
  --target CLOUD \
  --flow-actions-role-arn myRoleARN

```

Output:

```

{
  "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
  }
}

```

Per ulteriori informazioni, consulta [Working with Systems and Flow Configurations](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [CreateSystemInstance](#) in AWS CLI Command Reference.

## create-system-template

Il seguente esempio di codice mostra come utilizzare `create-system-template`.

### AWS CLI

Per creare un sistema

L'`create-system-template` esempio seguente crea un sistema. Il valore di `MySystemDefinition` è il GraphQL che modella il sistema.

```
aws iotthingsgraph create-system-template \  
  --definition language=GRAPHQL,text="MySystemDefinition"
```

Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, consulta [Creating Systems](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [CreateSystemTemplate](#) in AWS CLI Command Reference.

## delete-flow-template

Il seguente esempio di codice mostra come utilizzare `delete-flow-template`.

### AWS CLI

Per eliminare un flusso

L'`delete-flow-template` esempio seguente elimina un flusso (workflow).

```
aws iotthingsgraph delete-flow-template \  
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) nella AWS IoT Things Graph User Guide.

- Per i API dettagli, vedere [DeleteFlowTemplate](#) in Command Reference.AWS CLI

## delete-namespace

Il seguente esempio di codice mostra come utilizzare `delete-namespace`.

### AWS CLI

Per eliminare un namespace

L'`delete-namespace` esempio seguente elimina uno spazio dei nomi.

```
aws iotthingsgraph delete-namespace
```

Output:

```
{  
  "namespaceArn": "arn:aws:iotthingsgraph:us-west-2:123456789012",  
  "namespaceName": "us-west-2/123456789012/default"  
}
```

Per ulteriori informazioni, consulta [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) nella AWS IoT Things Graph User Guide.

- Per i API dettagli, vedere [DeleteNamespace](#) in Command Reference.AWS CLI

## delete-system-instance

Il seguente esempio di codice mostra come utilizzare `delete-system-instance`.

### AWS CLI

Per eliminare un'istanza di sistema

L'`delete-system-instance` seguente elimina un'istanza di sistema.

```
aws iotthingsgraph delete-system-instance \  
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) nella AWS IoT Things Graph User Guide.

- Per i API dettagli, vedere [DeleteSystemInstance](#) in Command Reference.AWS CLI

## `delete-system-template`

Il seguente esempio di codice mostra come utilizzare `delete-system-template`.

AWS CLI

Per eliminare un sistema

L'`delete-system-template` seguente elimina un sistema.

```
aws iotthingsgraph delete-system-template \  
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) nella AWS IoT Things Graph User Guide.

- Per i API dettagli, vedere [DeleteSystemTemplate](#) in Command Reference.AWS CLI

## `deploy-system-instance`

Il seguente esempio di codice mostra come utilizzare `deploy-system-instance`.

AWS CLI

Per distribuire un'istanza di sistema

L'`delete-system-template` seguente distribuisce un'istanza di sistema.

```
aws iotthingsgraph deploy-system-instance \  
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, consulta [Working with Systems and Flow Configurations](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [DeploySystemInstance](#) in AWS CLI Command Reference.

## deprecate-flow-template

Il seguente esempio di codice mostra come utilizzare `deprecate-flow-template`.

AWS CLI

Per deprecare un flusso

L'`deprecate-flow-template` esempio seguente rende obsoleto un flusso (workflow).

```
aws iotthingsgraph deprecate-flow-template \  
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) nella AWS IoT Things Graph User Guide.

- Per i API dettagli, vedere [DeprecateFlowTemplate](#) in Command Reference.AWS CLI

## deprecate-system-template

Il seguente esempio di codice mostra come utilizzare `deprecate-system-template`.

### AWS CLI

Deprecare un sistema

L'`deprecate-system-template` esempio seguente rende obsoleto un sistema.

```
aws iotthingsgraph deprecate-system-template \  
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) nella AWS IoT Things Graph User Guide.

- Per i API dettagli, vedere [DeprecateSystemTemplate](#) in Command Reference.AWS CLI

## describe-namespace

Il seguente esempio di codice mostra come utilizzare `describe-namespace`.

### AWS CLI

Per ottenere una descrizione del tuo namespace

L'`describe-namespace` esempio seguente ottiene una descrizione del tuo namespace.

```
aws iotthingsgraph describe-namespace
```

Output:

```
{  
  "namespaceName": "us-west-2/123456789012/default",  
  "trackingNamespaceName": "aws",  
  "trackingNamespaceVersion": 1,  
  "namespaceVersion": 5  
}
```

Per ulteriori informazioni, consulta [Namespace](#) nella Guida per l'utente di AWS IoT Things Graph.

- Per API i dettagli, vedere [DescribeNamespace](#) in AWS CLI Command Reference.

## dissociate-entity-from-thing

Il seguente esempio di codice mostra come utilizzare `dissociate-entity-from-thing`.

### AWS CLI

Dissociare un oggetto da un dispositivo

L'`dissociate-entity-from-thing` esempio seguente dissocia un oggetto da un dispositivo.

```
aws iotthingsgraph dissociate-entity-from-thing \  
  --thing-name "MotionSensorName" \  
  --entity-type "DEVICE"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creazione e caricamento di modelli](#) nella Guida per l'utente di AWS IoT Things Graph.

- Per API i dettagli, vedere [DissociateEntityFromThing](#) in AWS CLI Command Reference.

## get-entities

Il seguente esempio di codice mostra come utilizzare `get-entities`.

### AWS CLI

Per ottenere le definizioni delle entità

L'`get-entities` esempio seguente ottiene una definizione per un modello di dispositivo.

```
aws iotthingsgraph get-entities \  
  --ids "urn:tdm:aws/examples:DeviceModel:MotionSensor"
```

Output:

```
{
```

```

    "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}"
        }
      }
    ]
  }

```

Per ulteriori informazioni, consulta [Creazione e caricamento di modelli](#) nella Guida per l'utente di AWS IoT Things Graph.

- Per API i dettagli, vedere [GetEntities](#) in AWS CLI Command Reference.

## get-flow-template-revisions

Il seguente esempio di codice mostra come utilizzare `get-flow-template-revisions`.

### AWS CLI

Per ottenere informazioni di revisione su un flusso

L'`get-flow-template-revisions` seguente ottiene informazioni di revisione su un flusso (workflow).

```

aws iotthingsgraph get-flow-template-revisions \
  --id urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow

```

Output:

```

{
  "summaries": [
    {
      "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
      "revisionNumber": 1,
    }
  ]
}

```







## get-system-instance

Il seguente esempio di codice mostra come utilizzare `get-system-instance`.

### AWS CLI

Per ottenere un'istanza di sistema

L'`get-system-instance` esempio seguente ottiene una definizione per un'istanza di sistema.

```
aws iotthingsgraph get-system-instance \  
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

Output:

```
{  
  "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  }"}  
    },  
    "metricsConfiguration": {  
      "cloudMetricEnabled": false  
    },  
    "validatedNamespaceVersion": 5,  
    "flowActionsRoleArn": "arn:aws:iam::123456789012:role/ThingsGraphRole"  
  }  
}
```

```
}  
}
```

Per ulteriori informazioni, consulta [Working with Systems and Flow Configurations](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [GetSystemInstance](#) in AWS CLI Command Reference.

## get-system-template-revisions

Il seguente esempio di codice mostra come utilizzare `get-system-template-revisions`.

### AWS CLI

Per ottenere informazioni sulla revisione di un sistema

L'`get-system-template-revisions` esempio seguente ottiene informazioni sulla revisione di un sistema.

```
aws iotthingsgraph get-system-template-revisions \  
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

### Output:

```
{  
  "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  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Working with Systems and Flow Configurations](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [GetSystemTemplateRevisions](#) in AWS CLI Command Reference.

## get-system-template

Il seguente esempio di codice mostra come utilizzare `get-system-template`.

### AWS CLI

Per procurarsi un sistema

L'`get-system-template` esempio seguente ottiene una definizione per un sistema.

```
aws iotthingsgraph get-system-template \
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Working with Systems and Flow Configurations](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [GetSystemTemplate](#) in AWS CLI Command Reference.

## get-upload-status

Il seguente esempio di codice mostra come utilizzare `get-upload-status`.

### AWS CLI

Per conoscere lo stato della tua entità, carica

L'`get-upload-status` esempio seguente ottiene lo stato dell'operazione di caricamento dell'entità. Il valore di `MyUploadId` è il valore ID restituito dall'`upload-entity-definition` operazione.

```
aws iotthingsgraph get-upload-status \
  --upload-id "MyUploadId"
```

Output:

```
{
  "namespaceName": "us-west-2/123456789012/default",
  "namespaceVersion": 5,
  "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
  "uploadStatus": "SUCCEEDED"
}
```

Per ulteriori informazioni, consulta [Modeling Entities](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [GetUploadStatus](#) in AWS CLI Command Reference.

## list-flow-execution-messages

Il seguente esempio di codice mostra come utilizzare `list-flow-execution-messages`.

### AWS CLI

Per ottenere informazioni sugli eventi in un'esecuzione di flusso

L'`list-flow-execution-messages` esempio seguente ottiene informazioni sugli eventi in un'esecuzione di flusso.

```
aws iotthingsgraph list-flow-execution-messages \
```

```
--flow-execution-id "urn:tdm:us-west-2/123456789012/
default:Workflow:SecurityFlow_2019-05-11T19:39:55.317Z_MotionSensor_69b151ad-
a611-42f5-ac21-fe537f9868ad"
```

Output:

```
{
  "messages": [
    {
      "eventType": "EXECUTION_STARTED",
      "messageId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
      "payload": "Flow execution started",
      "timestamp": 1559247540.656
    }
  ]
}
```

Per ulteriori informazioni, consulta [Working with Flows](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [ListFlowExecutionMessages](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

AWS CLI

Per elencare tutti i tag di una risorsa

L'`list-tags-for-resource` esempio seguente elenca tutti i tag per una risorsa AWS IoT Things Graph.

```
aws iotthingsgraph list-tags-for-resource \
  --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
  default/Room218"
```

Output:

```
{
  "tags": [
    {
      "key": "Type",
```

```

        "value": "Residential"
      }
    ]
  }

```

Per ulteriori informazioni, consulta [Tagging Your AWS IoT Things Graph Resources](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## search-entities

Il seguente esempio di codice mostra come utilizzare `search-entities`.

### AWS CLI

Per cercare entità

L'esempio seguente cerca tutte le entità di tipo `EVENT`.

```

aws iotthingsgraph search-entities \
  --entity-types "EVENT"

```

Output:

```

{
  "descriptions": [
    {
      "id": "urn:tdm:aws/examples:Event:MotionSensorEvent",
      "type": "EVENT",
      "definition": {
        "language": "GRAPHQL",
        "text": "##\n# Description of events emitted by motion
sensor.\n##\nntype 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"
}

```

Per ulteriori informazioni, consulta [AWS IoT Things Graph Data Model Reference](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [SearchEntities](#) in AWS CLI Command Reference.

## search-flow-executions

Il seguente esempio di codice mostra come utilizzare `search-flow-executions`.

### AWS CLI

Per cercare le esecuzioni di flusso

L'`search-flow-executions` esempio seguente cerca tutte le esecuzioni di un flusso in un'istanza di sistema specificata.

```

aws iotthingsgraph search-flow-executions \
  --system-instance-id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"

```

Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta [Working with Systems and Flow Configurations](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [SearchFlowExecutions](#) in AWS CLI Command Reference.

## search-flow-templates

Il seguente esempio di codice mostra come utilizzare `search-flow-templates`.

### AWS CLI

Per cercare flussi (o flussi di lavoro)

L'`search-flow-templates` esempio seguente cerca tutti i flussi (flussi di lavoro) che contengono il modello di dispositivo Camera.

```
aws iotthingsgraph search-flow-templates \
  --filters name="DEVICE_MODEL_ID",value="urn:tdm:aws/examples:DeviceModel:Camera"
```

Output:

```
{
  "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
    }
  ]
}

```

Per ulteriori informazioni, consulta [Working with Flows](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [SearchFlowTemplates](#) in AWS CLI Command Reference.

## search-system-instances

Il seguente esempio di codice mostra come utilizzare `search-system-instances`.

### AWS CLI

Per cercare istanze di sistema

L'esempio seguente cerca tutte le istanze di sistema che contengono il sistema specificato.

```

aws iotthingsgraph search-system-instances \
  --filters name="SYSTEM_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/
  default:System:SecurityFlow"

```

Output:

```

{
  "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
  }
}
```

```
]
}
```

Per ulteriori informazioni, consulta [Working with Systems and Flow Configurations](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [SearchSystemInstances](#) in AWS CLI Command Reference.

## search-system-templates

Il seguente esempio di codice mostra come utilizzare `search-system-templates`.

### AWS CLI

Per cercare un sistema

L'esempio seguente cerca tutti i sistemi che contengono il flusso specificato.

```
aws iotthingsgraph search-system-templates \
  --filters name="FLOW_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/
  default:Workflow:SecurityFlow"
```

Output:

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta [Working with Flows](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [SearchSystemTemplates](#) in AWS CLI Command Reference.

## search-things

Il seguente esempio di codice mostra come utilizzare `search-things`.

### AWS CLI

Per cercare elementi associati a dispositivi e modelli di dispositivi

L'esempio seguente cerca tutti gli elementi associati al MotionSensor dispositivo HCSR5 01.

```
aws iotthingsgraph search-things \  
  --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Creazione e caricamento di modelli](#) nella Guida per l'utente di AWS IoT Things Graph.

- Per API i dettagli, vedere [SearchThings](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per creare un tag per una risorsa

L'esempio seguente crea un tag per la risorsa specificata.

```
aws iotthingsgraph tag-resource \  
  --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/  
default/Room218" \  
  --tags key="Type",value="Residential"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Your AWS IoT Things Graph Resources](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, consulta [TagResource AWS CLI Command Reference](#).

## undeploy-system-instance

Il seguente esempio di codice mostra come utilizzare `undeploy-system-instance`.

### AWS CLI

Per annullare la distribuzione di un'istanza di sistema dalla relativa destinazione

L'`undeploy-system-instance` seguente rimuove un'istanza di sistema dalla relativa destinazione.

```
aws iotthingsgraph undeploy-system-instance \  
  --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room215"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Lifecycle Management for AWS IoT Things Graph Entities, Flows, Systems and Deployments](#) nella AWS IoT Things Graph User Guide.

- Per i API dettagli, vedere [UndeploySystemInstance](#) in Command Reference.AWS CLI

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag per una risorsa

L'`untag-resource`esempio seguente rimuove un tag per la risorsa specificata.

```
aws iotthingsgraph untag-resource \  
  --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/  
default/Room218" \  
  --tag-keys "Type"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Your AWS IoT Things Graph Resources](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, consulta [UntagResource AWS CLI](#) Command Reference.

## update-flow-template

Il seguente esempio di codice mostra come utilizzare `update-flow-template`.

### AWS CLI

Per aggiornare un flusso

L'`update-flow-template`esempio seguente aggiorna un flusso (workflow). Il valore di `MyFlowDefinition` è il GraphQL che modella il flusso.

```
aws iotthingsgraph update-flow-template \  
  --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow" \  
  --definition language=GRAPHQL,text="MyFlowDefinition"
```



**Output:**

```
{
  "summary": {
    "createdAt": 1559248067.545,
    "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
    "revisionNumber": 2
  }
}
```

Per ulteriori informazioni, consulta [Working with Flows](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [UpdateFlowTemplate](#) in AWS CLI Command Reference.

**update-system-template**

Il seguente esempio di codice mostra come utilizzare `update-system-template`.

**AWS CLI**

Per aggiornare un sistema

L'`update-system-template` esempio seguente aggiorna un sistema. Il valore di `MySystemDefinition` è il GraphQL che modella il sistema.

```
aws iotthingsgraph update-system-template \
  --id "urn:tdm:us-west-2/123456789012/default:System:MySystem" \
  --definition language=GRAPHQL,text="MySystemDefinition"
```

**Output:**

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Creating Systems](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [UpdateSystemTemplate](#) in AWS CLI Command Reference.

## upload-entity-definitions

Il seguente esempio di codice mostra come utilizzare `upload-entity-definitions`.

### AWS CLI

Per caricare le definizioni delle entità

L'`upload-entity-definitions` seguente carica le definizioni di entità nel tuo namespace. Il valore di `MyEntityDefinitions` è il GraphQL che modella le entità.

```
aws iotthingsgraph upload-entity-definitions \  
  --document language=GRAPHQL,text="MyEntityDefinitions"
```

Output:

```
{  
  "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da"  
}
```

Per ulteriori informazioni, consulta [Modeling Entities](#) nella AWS IoT Things Graph User Guide.

- Per API i dettagli, vedere [UploadEntityDefinitions](#) in AWS CLI Command Reference.

## Wireless AWS IoT esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Wireless AWS IoT.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **associate-aws-account-with-partner-account**

Il seguente esempio di codice mostra come utilizzare `associate-aws-account-with-partner-account`.

#### AWS CLI

Per associare un account partner al tuo AWS account

L'esempio seguente associa le seguenti credenziali dell'account Sidewalk al tuo account. AWS

```
aws iotwireless associate-aws-account-with-partner-account \
  --sidewalk
  AmazonId="12345678901234",AppServerPrivateKey="a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
```

Output:

```
{
  "Sidewalk": {
    "AmazonId": "12345678901234",
    "AppServerPrivateKey":
    "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
  }
}
```

Per ulteriori informazioni, consulta [Amazon Sidewalk Integration for AWS IoT Core nella IoT Developers Guide](#).AWS

- Per API i dettagli, consulta [AssociateAwsAccountWithPartnerAccount AWS CLI Command Reference](#).

### **associate-wireless-device-with-thing**

Il seguente esempio di codice mostra come utilizzare `associate-wireless-device-with-thing`.

#### AWS CLI

Associare un oggetto a un dispositivo wireless

L'associazione `associate-wireless-device-with-thing` seguente associa un oggetto al dispositivo wireless con l'ID specificato.

```
aws iotwireless associate-wireless-device-with-thing \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
  --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere i gateway e i dispositivi wireless a AWS IoT Core LoRa WAN nella Guida per sviluppatori AWS IoT](#).

- Per API i dettagli, consulta [AssociateWirelessDeviceWithThing AWS CLI Command Reference](#).

## `associate-wireless-gateway-with-certificate`

Il seguente esempio di codice mostra come utilizzare `associate-wireless-gateway-with-certificate`.

### AWS CLI

Per associare il certificato al gateway wireless

Quanto segue `associate-wireless-gateway-with-certificate` associa un gateway wireless a un certificato.

```
aws iotwireless associate-wireless-gateway-with-certificate \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
  --iot-certificate-  
  id "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
```

Output:

```
{  
  "IotCertificateId":  
  "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"  
}
```

Per ulteriori informazioni, consulta [Aggiungere i gateway e i dispositivi wireless a AWS IoT Core LoRa WAN nella Guida per sviluppatori AWS IoT](#).

- Per API i dettagli, consulta [AssociateWirelessGatewayWithCertificate AWS CLI Command Reference](#).

## **associate-wireless-gateway-with-thing**

Il seguente esempio di codice mostra come utilizzare `associate-wireless-gateway-with-thing`.

### AWS CLI

Associare un oggetto a un gateway wireless

L'esempio seguente associa un oggetto a un gateway wireless.

```
aws iotwireless associate-wireless-gateway-with-thing \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
  --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere i gateway e i dispositivi wireless a AWS IoT Core LoRa WAN nella Guida per sviluppatori AWS IoT](#).

- Per API i dettagli, consulta [AssociateWirelessGatewayWithThing AWS CLI Command Reference](#).

## **create-destination**

Il seguente esempio di codice mostra come utilizzare `create-destination`.

### AWS CLI

Per creare una destinazione wireless IoT

L'esempio seguente crea una destinazione per mappare un messaggio del dispositivo a una regola AWS IoT. Prima di eseguire questo comando, è necessario aver creato un IAM ruolo che fornisca a AWS IoT Core LoRa WAN le autorizzazioni necessarie per inviare dati alla regola AWS IoT.

```
aws iotwireless create-destination \  
  --rule-arn "arn:aws:iot:us-east-1:123456789012:rule/MyIoTWirelessRule"
```

```
--name IoWirelessDestination \  
--expression-type RuleName \  
--expression IoWirelessRule \  
--role-arn arn:aws:iam::123456789012:role/IoWirelessDestinationRole
```

Output:

```
{  
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/  
IoWirelessDestination",  
  "Name": "IoWirelessDestination"  
}
```

Per ulteriori informazioni, consulta [Aggiungere destinazioni a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [CreateDestination AWS CLI](#) Command Reference.

## create-device-profile

Il seguente esempio di codice mostra come utilizzare `create-device-profile`.

### AWS CLI

Per creare un nuovo profilo del dispositivo

L'`create-device-profile` esempio seguente crea un nuovo profilo di dispositivo wireless IoT.

```
aws iotwireless create-device-profile
```

Output:

```
{  
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-  
a1b2-3c45-67d8-e90fa1b2c34d",  
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"  
}
```

Per ulteriori informazioni, consulta [Aggiungere profili a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [CreateDeviceProfile AWS CLI](#) Command Reference.

## create-service-profile

Il seguente esempio di codice mostra come utilizzare `create-service-profile`.

### AWS CLI

Per creare un nuovo profilo di servizio

L'esempio seguente crea un nuovo profilo di servizio wireless IoT.

```
aws iotwireless create-service-profile
```

Output:

```
{
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Per ulteriori informazioni, consulta [Aggiungere profili a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [CreateServiceProfile AWS CLI Command Reference](#).

## create-wireless-device

Il seguente esempio di codice mostra come utilizzare `create-wireless-device`.

### AWS CLI

Per creare un dispositivo wireless IoT

L'esempio seguente crea una risorsa di dispositivo wireless del tipo LoRaWAN.

```
aws iotwireless create-wireless-device \
  --cli-input-json file://input.json
```

Contenuto di `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
}
```

Output:

```
{
  "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f",
  "Id": "1ffd32c8-8130-4194-96df-622f072a315f"
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRaWAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [CreateWirelessDevice AWS CLI](#) Command Reference.

## create-wireless-gateway-task-definition

Il seguente esempio di codice mostra come utilizzare `create-wireless-gateway-task-definition`.

### AWS CLI

Per creare una definizione di attività per un gateway wireless

Quanto segue crea `create-wireless-gateway-task-definition` automaticamente le attività utilizzando questa definizione di attività per tutti i gateway con la versione corrente specificata.



```
aws iotwireless create-wireless-gateway-task-definition \  
--cli-input-json file://input.json
```

Contenuto di 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"  
      }  
    }  
  }  
}
```

Output:

```
{  
  "Id": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"  
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRaWAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [CreateWirelessGatewayTaskDefinition AWS CLI Command Reference](#).

## create-wireless-gateway-task

Il seguente esempio di codice mostra come utilizzare `create-wireless-gateway-task`.

## AWS CLI

Per creare l'attività per un gateway wireless

L'`create-wireless-gateway-task` seguente crea un'attività per un gateway wireless.

```
aws iotwireless create-wireless-gateway-task \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
  --wireless-gateway-task-definition-id "aa000102-0304-b0cd-ef56-a1b23cde456a"
```

Output:

```
{  
  "WirelessGatewayTaskDefinitionId": "aa204003-0604-30fb-ac82-a4f95aaf450a",  
  "Status": "Success"  
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [CreateWirelessGatewayTask AWS CLI Command Reference](#).

## `create-wireless-gateway`

Il seguente esempio di codice mostra come utilizzare `create-wireless-gateway`.

### AWS CLI

Per creare un gateway wireless

L'`create-wireless-gateway` seguente crea un gateway per LoRa WAN dispositivi wireless.

```
aws iotwireless create-wireless-gateway \  
  --lorawan GatewayEui="a1b2c3d4567890ab",RfRegion="US915" \  
  --name "myFirstLoRaWANGateway" \  
  --description "Using my first LoRaWAN gateway"
```

Output:

```
{
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [CreateWirelessGateway AWS CLI](#) Command Reference.

## delete-destination

Il seguente esempio di codice mostra come utilizzare `delete-destination`.

### AWS CLI

Per eliminare una destinazione wireless IoT

L'`delete-destination` esempio seguente elimina la risorsa di destinazione wireless con il nome `IoTWirelessDestination` che hai creato.

```
aws iotwireless delete-destination \
  --name "IoTWirelessDestination"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere destinazioni a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [DeleteDestination AWS CLI](#) Command Reference.

## delete-device-profile

Il seguente esempio di codice mostra come utilizzare `delete-device-profile`.

### AWS CLI

Per eliminare il profilo di un dispositivo

L'`delete-device-profile` esempio seguente elimina un profilo di dispositivo con l'ID specificato che avete creato.

```
aws iotwireless delete-device-profile \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere profili a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [DeleteDeviceProfile AWS CLI](#) Command Reference.

## delete-service-profile

Il seguente esempio di codice mostra come utilizzare `delete-service-profile`.

### AWS CLI

Per eliminare un profilo di servizio

L'`delete-service-profile` seguente elimina un profilo di servizio con l'ID specificato che avete creato.

```
aws iotwireless delete-service-profile \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere profili a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [DeleteServiceProfile AWS CLI](#) Command Reference.

## delete-wireless-device

Il seguente esempio di codice mostra come utilizzare `delete-wireless-device`.

### AWS CLI

Per eliminare un dispositivo wireless

L'`delete-wireless-device` seguente elimina un dispositivo wireless con l'ID specificato.

```
aws iotwireless delete-wireless-device \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [DeleteWirelessDevice AWS CLI Command Reference](#).

## delete-wireless-gateway-task-definition

Il seguente esempio di codice mostra come utilizzare `delete-wireless-gateway-task-definition`.

### AWS CLI

Per eliminare una definizione di attività su un gateway wireless

L'`delete-wireless-gateway-task-definition` seguente elimina la definizione di attività del gateway wireless creata con il seguente ID.

```
aws iotwireless delete-wireless-gateway-task-definition \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [DeleteWirelessGatewayTaskDefinition AWS CLI Command Reference](#).

## delete-wireless-gateway-task

Il seguente esempio di codice mostra come utilizzare `delete-wireless-gateway-task`.

### AWS CLI

Per eliminare un'operazione relativa al gateway wireless

L'`delete-wireless-gateway-task` seguente elimina l'attività del gateway wireless con l'ID specificato.

```
aws iotwireless delete-wireless-gateway-task \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [DeleteWirelessGatewayTask AWS CLI](#) Command Reference.

## `delete-wireless-gateway`

Il seguente esempio di codice mostra come utilizzare `delete-wireless-gateway`.

### AWS CLI

Per eliminare un gateway wireless

L'`delete-wireless-gateway` seguente elimina un gateway wireless con l'ID specificato.

```
aws iotwireless delete-wireless-gateway \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [DeleteWirelessGateway AWS CLI](#) Command Reference.

## `disassociate-aws-account-from-partner-account`

Il seguente esempio di codice mostra come utilizzare `disassociate-aws-account-from-partner-account`.

### AWS CLI

Per dissociare l'account partner dall' AWS account

L'`disassociate-aws-account-from-partner-account` seguente dissocia un account partner dall'account attualmente associato. AWS

```
aws iotwireless disassociate-aws-account-from-partner-account \  
  --partner-account-id "12345678901234" \  
  --partner-type "Sidewalk"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere i gateway e i dispositivi wireless a AWS IoT Core LoRa WAN nella Guida per sviluppatori AWS IoT](#).

- Per API i dettagli, consulta [DisassociateAwsAccountFromPartnerAccount AWS CLI Command Reference](#).

## `disassociate-wireless-device-from-thing`

Il seguente esempio di codice mostra come utilizzare `disassociate-wireless-device-from-thing`.

AWS CLI

Per dissociare l'oggetto dal dispositivo wireless

L'`disassociate-wireless-device-from-thing` seguente dissocia un dispositivo wireless dall'oggetto attualmente associato.

```
aws iotwireless disassociate-wireless-device-from-thing \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere i gateway e i dispositivi wireless a AWS IoT Core LoRa WAN nella Guida per sviluppatori AWS IoT](#).

- Per API i dettagli, consulta [DisassociateWirelessDeviceFromThing AWS CLI Command Reference](#).

## `disassociate-wireless-gateway-from-certificate`

Il seguente esempio di codice mostra come utilizzare `disassociate-wireless-gateway-from-certificate`.

## AWS CLI

Per dissociare il certificato dal gateway wireless

Quanto segue `disassociate-wireless-gateway-from-certificate` dissocia un gateway wireless dal certificato attualmente associato.

```
aws iotwireless disassociate-wireless-gateway-from-certificate \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere i gateway e i dispositivi wireless a AWS IoT Core LoRa WAN nella Guida per sviluppatori AWS IoT](#).

- Per API i dettagli, consulta [DisassociateWirelessGatewayFromCertificate AWS CLI Command Reference](#).

## `disassociate-wireless-gateway-from-thing`

Il seguente esempio di codice mostra come utilizzare `disassociate-wireless-gateway-from-thing`.

## AWS CLI

Per dissociare l'oggetto dal gateway wireless

L'`disassociate-wireless-gateway-from-thing` seguente dissocia un gateway wireless dall'elemento attualmente associato.

```
aws iotwireless disassociate-wireless-gateway-from-thing \  
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere i gateway e i dispositivi wireless a AWS IoT Core LoRa WAN nella Guida per sviluppatori AWS IoT](#).

- Per API i dettagli, consulta [DisassociateWirelessGatewayFromThing AWS CLI Command Reference](#).



## get-destination

Il seguente esempio di codice mostra come utilizzare `get-destination`.

### AWS CLI

Per ottenere informazioni su una destinazione wireless IoT

L'esempio seguente ottiene informazioni sulla risorsa di destinazione con il nome `IoTWirelessDestination` che hai creato.

```
aws iotwireless get-destination \  
  --name "IoTWirelessDestination"
```

Output:

```
{  
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/  
IoTWirelessDestination",  
  "Name": "IoTWirelessDestination",  
  "Expression": "IoTWirelessRule",  
  "ExpressionType": "RuleName",  
  "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"  
}
```

Per ulteriori informazioni, consulta [Aggiungere destinazioni a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [GetDestination AWS CLI Command Reference](#).

## get-device-profile

Il seguente esempio di codice mostra come utilizzare `get-device-profile`.

### AWS CLI

Per ottenere informazioni sul profilo di un dispositivo

L'esempio seguente ottiene informazioni sul profilo del dispositivo con l'ID specificato che hai creato.

```
aws iotwireless get-device-profile \  
  --device-profile-id "IoTWirelessDeviceProfile"
```

```
--id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Aggiungere profili a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [GetDeviceProfile AWS CLI](#) Command Reference.

## get-partner-account

Il seguente esempio di codice mostra come utilizzare `get-partner-account`.

### AWS CLI

Per ottenere le informazioni sull'account partner

L'`get-partner-account` esempio seguente ottiene informazioni sul tuo account Sidewalk con il seguente ID.

```
aws iotwireless get-partner-account \
  --partner-account-id "12345678901234" \
  --partner-type "Sidewalk"
```

Output:

```
{
  "Sidewalk": {
    "AmazonId": "12345678901234",
    "Fingerprint":
"a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
  },
  "AccountLinked": false
}
```

Per ulteriori informazioni, consulta [Amazon Sidewalk Integration for AWS IoT Core nella IoT Developers Guide.AWS](#)

- Per API i dettagli, consulta [GetPartnerAccount AWS CLI Command Reference](#).

## get-service-endpoint

Il seguente esempio di codice mostra come utilizzare `get-service-endpoint`.

### AWS CLI

Per ottenere l'endpoint del servizio

L'`get-service-endpoint` seguente ottiene l'endpoint specifico dell'account per il protocollo. CUPS

```
aws iotwireless get-service-endpoint
```

Output:

```
{
  "ServiceType": "CUPS",
  "ServiceEndpoint": "https://A1RMKZ37ACAGOT.cups.lorawan.us-east-1.amazonaws.com:443",
  "ServerTrust": "-----BEGIN CERTIFICATE-----\n
MIIESTCCAzGgAwIBAgITBn+UV4WH6Kx33rJTMlu8mYtWDTANBgkqhkiG9w0BAQsF\n
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6\n
b24gUm9vdCBDQSAxMB4XDTE1MTAyMjAwMDAwMFoXDTE1MTAxOTAwMDAwMFowRjEL\n
MAKGA1UEBhMCVVMxMzANBgNVBAoTBkFtYXN0YXN0EjEwMC4GA1UEAxMwYXN0YXN0\n
IDFCMQ8wDQYDVQQDEwZBbWF6b24wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK\n
AoIBAQCThZn3c68asg3Wuw6MLAd5tES6BIOsMzoKcG5b1PVo+sD0RrMd4f2AbnZ\n
cMzPa43j4wNxp1ty6aUKk4T1qe9B0wKFjwK6zmxLlVYo7bHViXsP1J6q0MpFge5\n
b1DP+18x+B26A0piiQ0uPkfyDyeR4xQghfj66Yo19V+emU3nazfvpFA+R0z6WoVm\n
-----"
```

```

B5x+F2pV8xeKNR7u6azDdU5YVX1Tawp1mxRC1+WsAYmz6qP+z8ArDITC2FMVy2fw\n
0IjK0tEXc/VfmtTFch5+AfGYMGmqqvJ6LcXiAhqG5TI+Dr0RtM88k+8XUBCeQ8IG\n
KuANaL7TiItKZYxK1MMuTJtV9Ib1AgMBAAGjggE7MIIBNzASBgNVHRMBAf8ECDAG\n
AQH/AgEAMA4GA1UdDwEB/wQEAwIBhjAdBgNVHQ4EFgQUWaRmBlKge5WSPK0UByeW\n
dFv5PdAwHwYDVR0jBBgwFoAUhBjMhTTsvAyU1C4IWZzHshB0CggwewYIKwYBBQUH\n
AQEEbzBtMC8GCCsGAQUFBzABhiNodHRwOi8vb2NzcC5yb290Y2ExLmFtYXpvbnRy\n
dXN0LmNvbTA6BggrBgEFBQcwAoYuaHR0cDovL2Nydc5yb290Y2ExLmFtYXpvbnRy\n
dXN0LmNvbS9yb290Y2ExLmNlclcA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3Js\n
LnJvb3RjYTEuYW1hem9udHJ1c3QuY29tL3Jvb3RjYTEuY3JsMBMGA1UdIAQMMMAow\n
CAYGZ4EMAQIBMA0GCSqGSIb3DQEBCwUAA4IBAQCfkr41u3nPo4FCH0TjY3NT0VI1\n
59Gt/a6ZiqyJEi+752+a1U5y6iAwYfmXss21JwJFqMp2PphKg5625kXg8kP2CN5t\n
6G7bMQcT8C8xDZntYTd7WPD8UZiRKAJPBXa30/AbwuZe0GaFEQ8ugcYQgSn+IGBI\n
8/LwhBNTZTUVEWuCUUBVV18YtbAiPq3yXqMB480z+ctBWuZSkbvkNodPLamkBg1\n
upRyzQ7qDn1X8nn8N8V7YJ6y68AtkHcNSRAnpTitxBKjtkPISLMVCx7i4hncxHZS\n
yLyKQXhw2W2Xs0qLeC1etA+jTGDK4UfLeC0SF7FSi8o5LL21L8IzApar2pR/\n
-----END CERTIFICATE-----\n"
}

```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [GetServiceEndpoint AWS CLI Command Reference](#).

## get-service-profile

Il seguente esempio di codice mostra come utilizzare `get-service-profile`.

### AWS CLI

Per ottenere informazioni su un profilo di servizio

L'`get-service-profile` esempio seguente ottiene informazioni sul profilo di servizio con l'ID specificato che hai creato.

```

aws iotwireless get-service-profile \
  --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"

```

Output:

```

{
  "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
}
```

Per ulteriori informazioni, consulta [Aggiungere profili a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [GetServiceProfile AWS CLI Command Reference](#).

## get-wireless-device-statistics

Il seguente esempio di codice mostra come utilizzare `get-wireless-device-statistics`.

### AWS CLI

Per ottenere informazioni operative su un dispositivo wireless

L'esempio seguente ottiene informazioni operative su un dispositivo wireless.

```
aws iotwireless get-wireless-device-statistics \
  --wireless-device-id "1ffd32c8-8130-4194-96df-622f072a315f"
```

Output:

```
{
  "WirelessDeviceId": "1ffd32c8-8130-4194-96df-622f072a315f"
```

```
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [GetWirelessDeviceStatistics AWS CLI](#) Command Reference.

## get-wireless-device

Il seguente esempio di codice mostra come utilizzare `get-wireless-device`.

### AWS CLI

Per ottenere informazioni sul dispositivo wireless

L'`get-wireless-device` esempio seguente elenca i widget disponibili nel tuo AWS account.

```
aws iotwireless get-wireless-device \  
  --identifier "1ffd32c8-8130-4194-96df-622f072a315f" \  
  --identifier-type WirelessDeviceID
```

Output:

```
{  
  "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/1ffd32c8-8130-4194-96df-622f072a315f",
```

```
"Description": "My LoRaWAN wireless device"
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRaWAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [GetWirelessDevice AWS CLI](#) Command Reference.

## get-wireless-gateway-certificate

Il seguente esempio di codice mostra come utilizzare `get-wireless-gateway-certificate`.

### AWS CLI

Per ottenere l'ID di un certificato associato a un gateway wireless

L'`get-wireless-gateway-certificate` esempio seguente ottiene l'ID del certificato associato a un gateway wireless con l'ID specificato.

```
aws iotwireless get-wireless-gateway-certificate \
  --id "6c44ab31-8b4d-407a-bed3-19b6c7cda551"
```

Output:

```
{
  "IotCertificateId":
  "8ea4aeae3db34c78cce75d9abd830356869ead6972997e0603e5fd032c804b6f"
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRaWAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [GetWirelessGatewayCertificate AWS CLI](#) Command Reference.

## get-wireless-gateway-firmware-information

Il seguente esempio di codice mostra come utilizzare `get-wireless-gateway-firmware-information`.

### AWS CLI

Per ottenere informazioni sul firmware relative a un gateway wireless

L'get-wireless-gateway-firmware-informationesempio seguente ottiene la versione del firmware e altre informazioni su un gateway wireless.

```
aws iotwireless get-wireless-gateway-firmware-information \  
--id "3039b406-5cc9-4307-925b-9948c63da25b"
```

Output:

```
{  
  "LoRaWAN" :{  
    "CurrentVersion" :{  
      "PackageVersion" : "1.0.0",  
      "Station" : "2.0.5",  
      "Model" : "linux"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRaWAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [GetWirelessGatewayFirmwareInformation AWS CLI Command Reference](#).

## get-wireless-gateway-statistics

Il seguente esempio di codice mostra come utilizzareget-wireless-gateway-statistics.

AWS CLI

Per ottenere informazioni operative su un gateway wireless

L'get-wireless-gateway-statisticsesempio seguente ottiene informazioni operative su un gateway wireless.

```
aws iotwireless get-wireless-gateway-statistics \  
--wireless-gateway-id "3039b406-5cc9-4307-925b-9948c63da25b"
```

Output:

```
{  
  "WirelessGatewayId": "3039b406-5cc9-4307-925b-9948c63da25b"
```



```
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [GetWirelessGatewayStatistics AWS CLI](#) Command Reference.

## get-wireless-gateway-task-definition

Il seguente esempio di codice mostra come utilizzare `get-wireless-gateway-task-definition`.

### AWS CLI

Per ottenere informazioni su un gateway wireless, definizione delle attività

L'`get-wireless-gateway-task-definition` esempio seguente ottiene informazioni sulla definizione dell'attività wireless con l'ID specificato.

```
aws iotwireless get-wireless-gateway-task-definition \  
  --id "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"
```

Output:

```
{  
  "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"  
      }  
    }  
  }  
}
```

```
}  
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [GetWirelessGatewayTaskDefinition AWS CLI](#) Command Reference.

## get-wireless-gateway-task

Il seguente esempio di codice mostra come utilizzare `get-wireless-gateway-task`.

### AWS CLI

Per ottenere informazioni sull'operazione relativa al gateway wireless

L'esempio seguente ottiene informazioni sull'attività del gateway wireless con l'ID specificato.

```
aws iotwireless get-wireless-gateway-task \  
  --id "11693a46-6866-47c3-a031-c9a616e7644b"
```

Output:

```
{  
  "WirelessGatewayId": "6c44ab31-8b4d-407a-bed3-19b6c7cda551",  
  "WirelessGatewayTaskDefinitionId": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9",  
  "Status": "Success"  
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [GetWirelessGatewayTask AWS CLI](#) Command Reference.

## get-wireless-gateway

Il seguente esempio di codice mostra come utilizzare `get-wireless-gateway`.

### AWS CLI

Per ottenere informazioni su un gateway wireless

L'get-wireless-gatewayesempio seguente ottiene informazioni sul gateway wirelessmyFirstLoRaWANGateway.

```
aws iotwireless get-wireless-gateway \
  --identifier "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
  --identifier-type WirelessGatewayId
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRaWAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [GetWirelessGateway AWS CLI](#) Command Reference.

## list-destinations

Il seguente esempio di codice mostra come utilizzarelist-destinations.

AWS CLI

Per elencare le destinazioni wireless

L'list-destinationesempio seguente elenca le destinazioni disponibili registrate nel tuo AWS account.

```
aws iotwireless list-destinations
```

**Output:**

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Aggiungere destinazioni a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [ListDestinations AWS CLI](#) Command Reference.

**list-device-profiles**

Il seguente esempio di codice mostra come utilizzare `list-device-profiles`.

**AWS CLI**

Per elencare i profili dei dispositivi

L'`list-device-profiles` esempio seguente elenca i profili di dispositivo disponibili registrati nell' AWS account.

```
aws iotwireless list-device-profiles
```

**Output:**

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Aggiungere profili a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [ListDeviceProfiles AWS CLI Command Reference](#).

## list-partner-accounts

Il seguente esempio di codice mostra come utilizzare `list-partner-accounts`.

### AWS CLI

Per elencare gli account dei partner

L'`list-partner-accounts` esempio seguente elenca gli account partner disponibili associati al tuo AWS account.

```
aws iotwireless list-partner-accounts
```

Output:

```
{
  "Sidewalk": [
    {
      "AmazonId": "78965678771228",
      "Fingerprint":
        "bd96d8ef66dbfd2160eb60e156849e82ad7018b8b73c1ba0b4fc65c32498ee35"
    }
  ]
}
```

```
    },
    {
      "AmazonId": "89656787651228",
      "Fingerprint":
"bc5e99e151c07be14be7e6603e4489c53f858b271213a36ebe3370777ba06e9b"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Amazon Sidewalk Integration for AWS IoT Core nella IoT Developers Guide.AWS](#)

- Per API i dettagli, consulta [ListPartnerAccounts AWS CLI Command Reference](#).

## list-service-profiles

Il seguente esempio di codice mostra come utilizzare `list-service-profiles`.

### AWS CLI

Per elencare i profili di servizio

L'`list-service-profiles` esempio seguente elenca i profili di servizio disponibili registrati nell'AWS account dell'utente.

```
aws iotwireless list-service-profiles
```

Output:

```
{
  "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"
    }
  ]
}
```

```
}
```

Per ulteriori informazioni, consulta [Aggiungere profili a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [ListServiceProfiles AWS CLI Command Reference](#).

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag assegnati alla risorsa

L'`list-tags-for-resource` esempio seguente elenca i tag assegnati a una risorsa di destinazione wireless.

```
aws iotwireless list-tags-for-resource \  
  --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/  
  IoWirelessDestination"
```

Output:

```
{  
  "Tags": [  
    {  
      "Value": "MyValue",  
      "Key": "MyTag"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Descrivi il tuo AWS IoT Core per LoRa WAN le risorse](#) nella Guida per sviluppatori AWS IoT.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## list-wireless-devices

Il seguente esempio di codice mostra come utilizzare `list-wireless-devices`.

## AWS CLI

Per elencare i dispositivi wireless disponibili

L'`list-wireless-devices` seguente elenca i dispositivi wireless disponibili registrati nell' AWS account.

```
aws iotwireless list-wireless-devices
```

Output:

```
{
  "WirelessDeviceList": [
    {
      "Name": "myLoRaWANDevice",
      "DestinationName": "IoTWirelessDestination",
      "Id": "1ffd32c8-8130-4194-96df-622f072a315f",
      "Type": "LoRaWAN",
      "LoRaWAN": {
        "DevEui": "ac12efc654d23fc2"
      },
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRaWAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [ListWirelessDevices AWS CLI](#) Command Reference.

## `list-wireless-gateway-task-definitions`

Il seguente esempio di codice mostra come utilizzare `list-wireless-gateway-task-definitions`.

## AWS CLI

Per elencare le definizioni delle attività del gateway wireless



L'`list-wireless-gateway-task-definitions` seguente elenca le definizioni delle attività del gateway wireless disponibili registrate nell' AWS account.

```
aws iotwireless list-wireless-gateway-task-definitions
```

Output:

```
{
  "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"
          }
        }
    }
  ]
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRaWAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [ListWirelessGatewayTaskDefinitions AWS CLI Command Reference](#).

## **list-wireless-gateways**

Il seguente esempio di codice mostra come utilizzare `list-wireless-gateways`.

AWS CLI

Per elencare i gateway wireless

L'`list-wireless-gateways` seguente elenca i gateway wireless disponibili nell'account AWS .

```
aws iotwireless list-wireless-gateways
```

Output:

```
{
  "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": "cda123fffe92ecd2"
      },
      "Id": "3285bdc7-5a12-4991-84ed-dadca65e342e",
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/3285bdc7-5a12-4991-84ed-dadca65e342e",
      "Name": "mySecondLoRaWANGateway"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRaWAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [ListWirelessGateways AWS CLI Command Reference](#).

## send-data-to-wireless-device

Il seguente esempio di codice mostra come utilizzare `send-data-to-wireless-device`.

## AWS CLI

Per inviare dati al dispositivo wireless

L'example seguente invia un frame di dati dell'applicazione decrittografato al dispositivo wireless.

```
aws iotwireless send-data-to-wireless-device \  
  --id "11aa5eae-2f56-4b8e-a023-b28d98494e49" \  
  --transmit-mode "1" \  
  --payload-data "SGVsbG8gVG8gRGV2c2lt" \  
  --wireless-metadata LoRaWAN={FPort=1}
```

Output:

```
{  
  MessageId: "6011dd36-0043d6eb-0072-0008"  
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRaWAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [SendDataToWirelessDevice AWS CLI Command Reference](#).

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per specificare una chiave e un valore di tag per una risorsa

L'example seguente contrassegna la destinazione wireless `IoWirelessDestination` con la chiave `MyTag` e il valore `MyValue`.

```
aws iotwireless tag-resource \  
  --resource-arn "arn:aws:iotwireless:us-east-1:651419225604:Destination/  
IoWirelessDestination" \  
  --tags Key="MyTag",Value="MyValue"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Descrivi il tuo AWS IoT Core per LoRa WAN le risorse](#) nella Guida per sviluppatori AWS IoT.

- Per API i dettagli, consulta [TagResource AWS CLI](#) Command Reference.

## test-wireless-device

Il seguente esempio di codice mostra come utilizzare `test-wireless-device`.

### AWS CLI

Per testare il dispositivo wireless

L'`test-wireless-device` esempio seguente invia i dati di uplink Hello a un dispositivo con un ID specificato.

```
aws iotwireless test-wireless-device \  
  --id "11aa5eae-2f56-4b8e-a023-b28d98494e49"
```

Output:

```
{  
  Result: "Test succeeded. one message is sent with payload: hello"  
}
```

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [TestWirelessDevice AWS CLI](#) Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere uno o più tag da una risorsa

L'`untag-resource` esempio seguente rimuove il tag `MyTag` e il relativo valore dalla destinazione `wirelessIoTWirelessDestination`.

```
aws iotwireless untag-resource \  
  --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/  
IoWirelessDestination" \  
  --tag-keys "MyTag"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Descrivi il tuo AWS IoT Core per LoRa WAN le risorse](#) nella Guida per sviluppatori AWS IoT.

- Per API i dettagli, consulta [UntagResource AWS CLI](#) Command Reference.

## update-destination

Il seguente esempio di codice mostra come utilizzare `update-destination`.

### AWS CLI

Per aggiornare le proprietà di una destinazione

L'`update-destination` esempio seguente aggiorna la proprietà di descrizione di una destinazione wireless.

```
aws iotwireless update-destination \  
  --name "IoWirelessDestination" \  
  --description "Destination for messages processed using IoWirelessRule"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere destinazioni a AWS IoT Core LoRa WAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [UpdateDestination AWS CLI](#) Command Reference.

## update-partner-account

Il seguente esempio di codice mostra come utilizzare `update-partner-account`.

### AWS CLI

Per aggiornare le proprietà di un account partner

Quanto segue `update-partner-account` aggiorna il `AppServerPrivateKey` per l'account con l'ID specificato.

```
aws iotwireless update-partner-account \  
  --partner-account-id "78965678771228" \  
  --partner-type "Sidewalk" \  
  --sidewalk  
  AppServerPrivateKey="f798ab4899346a88599180fee9e14fa1ada7b6df989425b7c6d2146dd6c815bb"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Amazon Sidewalk Integration for AWS IoT Core nella IoT Developers Guide](#).AWS

- Per API i dettagli, consulta [UpdatePartnerAccount AWS CLI Command Reference](#).

## update-wireless-device

Il seguente esempio di codice mostra come utilizzare `update-wireless-device`.

### AWS CLI

Per aggiornare le proprietà di un dispositivo wireless

L'`update-wireless-device`esempio seguente aggiorna le proprietà di un dispositivo wireless registrato nell' AWS account dell'utente.

```
aws iotwireless update-wireless-device \  
  --id "1ffd32c8-8130-4194-96df-622f072a315f" \  
  --destination-name IoTWirelessDestination2 \  
  --description "Using my first LoRaWAN device"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRaWAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [UpdateWirelessDevice AWS CLI Command Reference](#).

## update-wireless-gateway

Il seguente esempio di codice mostra come utilizzare `update-wireless-gateway`.

## AWS CLI

Per aggiornare il gateway wireless

L'update-wireless-gatewayesempio seguente aggiorna la descrizione del gateway wireless.

```
aws iotwireless update-wireless-gateway \  
  --id "3285bdc7-5a12-4991-84ed-dadca65e342e" \  
  --description "Using my LoRaWAN gateway"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Connessione di dispositivi e gateway a AWS IoT Core LoRaWAN](#) nella AWS IoT Developers Guide.

- Per API i dettagli, consulta [UpdateWirelessGateway AWS CLI](#) Command Reference.

## IVSEsempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con AmazonIVS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### batch-get-channel

Il seguente esempio di codice mostra come utilizzarebatch-get-channel.

AWS CLI

Per ottenere informazioni sulla configurazione dei canali su più canali

L'batch-get-channel seguente elenca le informazioni sui canali specificati.

```
aws ivs batch-get-channel \  
  --arns arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
         arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl
```

Output:

```
{  
  "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":  
"AB1C2defGHijklLMNo3PqQRstUvwxyzABCDefghh4ijklMN5opqrStuVWXYZAbCDEfghIJ"  
      },  
      "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":
"BA1C2defGHijkLMNo3PqQRstUvwxyzABCDefghh4ijk1MN5opqrStuVWxyzAbCDefghIJ"
        },
        "tags": {},
        "type": "STANDARD"
    }
]
}

```

Per ulteriori informazioni, vedete [Creare un canale nella Guida](#) per l'utente IVS a bassa latenza.

- Per API i dettagli, vedere [BatchGetChannel](#) in AWS CLI Command Reference.

## batch-get-stream-key

Il seguente esempio di codice mostra come utilizzare `batch-get-stream-key`.

### AWS CLI

Per ottenere informazioni su più chiavi di streaming

L'`batch-get-stream-key` seguente ottiene informazioni sulle chiavi di stream specificate.

```

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

```

Output:

```

{
  "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": {}
      }
    ]
  }

```

Per ulteriori informazioni, consultate [Create a Channel](#) nella IVS Low-Latency User Guide.

- Per API i dettagli, vedere [BatchGetStreamKey](#) in AWS CLI Command Reference.

## batch-start-viewer-session-revocation

Il seguente esempio di codice mostra come utilizzare `batch-start-viewer-session-revocation`.

### AWS CLI

Per revocare le sessioni di visualizzazione per più coppie di ID di canale ARN e visualizzatore

L'esempio seguente esegue la revoca della sessione su più coppie di canali e ID visualizzatore contemporaneamente. ARN La richiesta può essere completata normalmente, ma restituisce valori nel campo degli errori se il chiamante non è autorizzato a revocare la sessione specificata.

```

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}]'

```

Output:

```

{
  "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",
    }
  ]
}
```

Per maggiori informazioni, consulta [Impostazione dei canali privati](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, vedere [BatchStartViewerSessionRevocation](#) in AWS CLI Command Reference.

## create-channel

Il seguente esempio di codice mostra come utilizzare `create-channel`.

### AWS CLI

Esempio 1: creare un canale senza registrazione

L'`create-channel` esempio seguente crea un nuovo canale e una chiave di streaming associata per avviare lo streaming.

```
aws ivs create-channel \
  --name "test-channel" \
  --no-insecure-ingest
```

Output:

```
{
  "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": {}
}
}

```

Per ulteriori informazioni, consulta [Creare un canale nella Guida](#) per l'utente IVS a bassa latenza.

Esempio 2: per creare un canale con la registrazione abilitata, utilizzando la RecordingConfiguration risorsa specificata dal relativo ARN

L'create-channel esempio seguente crea un nuovo canale e una chiave di streaming associata per avviare lo streaming e imposta la registrazione per il canale.

```

aws ivs create-channel \
  --name test-channel-with-recording \
  --insecure-ingest \
  --recording-configuration-arn "arn:aws:ivs:us-west-2:123456789012:recording-
configuration/ABCD12cdEFgh"

```

Output:

```

{
  "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":
"BA1C2defGHijklMNop3PqQRstUvwxyzABCDefghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "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": {}
  }
}

```

Per ulteriori informazioni, consulta [Record to Amazon S3](#) nella IVSLow-Latency User Guide.

Esempio 3: per creare un canale con una politica di restrizione della riproduzione specificata dal ARN

L'create-channel esempio seguente crea un nuovo canale e una chiave di streaming associata per avviare lo streaming e imposta una politica di restrizione della riproduzione per il canale.

```

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"

```

Output:

```

{
  "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":
"AB1C2edfGHijklMNop3PqQRstUvwxyzABCDefghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "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": {}
  }
}

```

Per ulteriori informazioni, consulta [Contenuti e visualizzatori indesiderati nella Guida per l'IVSutente a bassa latenza](#).

- Per i API dettagli, vedere [CreateChannel](#) in Command Reference.AWS CLI

## create-playback-restriction-policy

Il seguente esempio di codice mostra come utilizzare `create-playback-restriction-policy`.

### AWS CLI

Per creare una politica di restrizione della riproduzione

L'esempio seguente crea una nuova politica di restrizione della riproduzione.

```
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
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Contenuti e visualizzatori indesiderati](#) nella Guida per l'utente a bassa latenza. IVS

- Per i API dettagli, vedere [CreatePlaybackRestrictionPolicy](#) in Command Reference.AWS CLI

## create-recording-configuration

Il seguente esempio di codice mostra come utilizzare `create-recording-configuration`.

AWS CLI

Per creare una `RecordingConfiguration` risorsa

L'create-recording-configurationesempio seguente crea una RecordingConfiguration risorsa per abilitare la registrazione su Amazon S3.

```
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}
```

Output:

```
{
  "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"
      ]
    }
  }
}
```



```

    ]
  }
}
}

```

Per ulteriori informazioni, consulta [Record to Amazon S3](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [CreateRecordingConfiguration AWS CLI Command Reference](#).

## create-stream-key

Il seguente esempio di codice mostra come utilizzare `create-stream-key`.

### AWS CLI

Per creare una chiave di streaming

L'esempio seguente crea una chiave di flusso per uno specifico ARN (Amazon Resource Name).

```

aws ivs create-stream-key \
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh

```

Output:

```

{
  "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": {}
  }
}

```

Per ulteriori informazioni, consulta [Create a Channel](#) nella IVS Low-Latency User Guide.

- Per API i dettagli, vedere [CreateStreamKey](#) in AWS CLI Command Reference.

## delete-channel

Il seguente esempio di codice mostra come utilizzare `delete-channel`.

## AWS CLI

Per eliminare un canale e le chiavi di streaming associate

L'`delete-channel` esempio seguente elimina il canale con il valore specificato ARN (Amazon Resource Name).

```
aws ivs delete-channel \  
  --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Create a Channel](#) nella IVS Low-Latency User Guide.

- Per API i dettagli, vedere [DeleteChannel](#) in AWS CLI Command Reference.

## `delete-playback-key-pair`

Il seguente esempio di codice mostra come utilizzare `delete-playback-key-pair`.

## AWS CLI

Per eliminare una coppia di key pair di riproduzione specificata

L'`delete-playback-key-pair` esempio seguente restituisce l'impronta digitale della coppia di chiavi specificata.

```
aws ivs delete-playback-key-pair \  
  --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh
```

Questo comando non produce alcun output.

Per maggiori informazioni, consulta [Impostazione dei canali privati](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, vedere [DeletePlaybackKeyPair](#) in AWS CLI Command Reference.

## `delete-playback-restriction-policy`

Il seguente esempio di codice mostra come utilizzare `delete-playback-restriction-policy`.

## AWS CLI

Per eliminare un criterio di restrizione della riproduzione

L'`delete-playback-restriction-policy` esempio seguente elimina la politica di restrizione della riproduzione con la politica specificata (ARN Amazon Resource Name).

```
aws ivs delete-playback-restriction-policy \  
  --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/  
  ABCdef34ghIJ"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Undesired Content and Viewers](#) nella Low-Latency User Guide. IVS

- Per i API dettagli, vedere [DeletePlaybackRestrictionPolicy](#) in Command Reference. AWS CLI

## `delete-recording-configuration`

Il seguente esempio di codice mostra come utilizzare `delete-recording-configuration`.

## AWS CLI

Per eliminare la `RecordingConfiguration` risorsa specificata dalla sua ARN

L'`delete-recording-configuration` esempio seguente elimina la `RecordingConfiguration` risorsa con il valore specificato ARN.

```
aws ivs delete-recording-configuration \  
  --arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABCdef34ghIJ"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Record to Amazon S3](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [DeleteRecordingConfiguration AWS CLI](#) Command Reference.

## `delete-stream-key`

Il seguente esempio di codice mostra come utilizzare `delete-stream-key`.

## AWS CLI

Per eliminare una chiave di streaming

L'`delete-stream-key` seguente elimina la chiave stream per uno specifico ARN (Amazon Resource Name), in modo che non possa più essere utilizzata per lo streaming.

```
aws ivs delete-stream-key \  
  --arn arn:aws:ivs:us-west-2:123456789012:stream-key/g1H2I3j4k5L6
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Create a Channel](#) nella IVSLow-Latency User Guide.

- Per API i dettagli, vedere [DeleteStreamKey](#) in AWS CLI Command Reference.

## get-channel

Il seguente esempio di codice mostra come utilizzare `get-channel`.

## AWS CLI

Per ottenere informazioni sulla configurazione di un canale

L'`get-channel` seguente ottiene la configurazione del canale per un canale specificato ARN (Amazon Resource Name).

```
aws ivs get-channel \  
  --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Output:

```
{  
  "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":
"AB1C2defGHijklMN03PqQRstUvwxyzABCDefghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "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": {}
  }
}

```

Per ulteriori informazioni, consulta [Create a Channel](#) nella IVS Low-Latency User Guide.

- Per API i dettagli, vedere [GetChannel](#) in AWS CLI Command Reference.

## get-playback-key-pair

Il seguente esempio di codice mostra come utilizzare `get-playback-key-pair`.

### AWS CLI

Per ottenere una coppia di key pair di riproduzione specificata

L'esempio seguente restituisce l'impronta digitale della coppia di chiavi specificata.

```

aws ivs get-playback-key-pair \
  --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh

```

Output:

```

{
  "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": {}
  }
}

```

Per maggiori informazioni, consulta [Impostazione dei canali privati](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, vedere [GetPlaybackKeyPair](#) in AWS CLI Command Reference.

## get-playback-restriction-policy

Il seguente esempio di codice mostra come utilizzare `get-playback-restriction-policy`.

### AWS CLI

Per ottenere le informazioni di configurazione di una politica di restrizione della riproduzione

L'`get-playback-restriction-policy` esempio seguente ottiene la configurazione della politica di restrizione della riproduzione con la politica specificata (ARN Amazon Resource Name).

```
aws ivs get-playback-restriction-policy \  
  --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/  
  ABCdef34ghIJ"
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Undesired Content and Viewers](#) nella Low-Latency User Guide. IVS

- Per i API dettagli, vedere [GetPlaybackRestrictionPolicy](#) in Command Reference.AWS CLI

## get-recording-configuration

Il seguente esempio di codice mostra come utilizzare `get-recording-configuration`.

### AWS CLI

Per ottenere informazioni su una RecordingConfiguration risorsa

L'`get-recording-configuration` esempio seguente ottiene informazioni sulla RecordingConfiguration risorsa per l'oggetto specificato ARN.

```
aws ivs get-recording-configuration \
  --arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABcdef34ghIJ"
```

Output:

```
{
  "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"
    ]
  }
}
}

```

Per ulteriori informazioni, consulta [Record to Amazon S3](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [GetRecordingConfiguration AWS CLI Command Reference](#).

## get-stream-key

Il seguente esempio di codice mostra come utilizzare `get-stream-key`.

### AWS CLI

Per ottenere informazioni su uno stream

L'`get-stream-key` esempio seguente ottiene informazioni sulla chiave di stream specificata.

```

aws ivs get-stream-key \
  --arn arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh --region=us-
west-2

```

Output:

```

{
  "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": {}
  }
}

```

Per ulteriori informazioni, consultate [Create a Channel](#) nella IVSLow-Latency User Guide.



- Per API i dettagli, vedere [GetStreamKey](#) in AWS CLI Command Reference.

## get-stream-session

Il seguente esempio di codice mostra come utilizzare `get-stream-session`.

### AWS CLI

Per ottenere i metadati per uno stream specifico

L'`get-stream-session` seguente ottiene la configurazione dei metadati per il canale specificato ARN (Amazon Resource Name) e lo stream specificato; se non `streamId` viene fornita, viene selezionato lo stream più recente per il canale.

```
aws ivs get-stream-session \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --stream-id "mystream"
```

Output:

```
{  
  "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":  
"AB1C2defGHijklMNop3PqRstUvwxyzABCDefghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"  
      },  
      "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"
    }
  ]
}
}

```

Per ulteriori informazioni, consulta [Create a Channel](#) nella IVSLow-Latency User Guide.

- Per API i dettagli, vedere [GetStreamSession](#) in AWS CLI Command Reference.

## get-stream

Il seguente esempio di codice mostra come utilizzare `get-stream`.

### AWS CLI

Per ottenere informazioni su uno stream

L'`get-stream` esempio seguente ottiene informazioni sullo stream per il canale specificato.

```
aws ivs get-stream \
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, vedete [Creare un canale nella Guida](#) per l'utente IVS a bassa latenza.

- Per API i dettagli, vedere [GetStream](#) in AWS CLI Command Reference.

## import-playback-key-pair

Il seguente esempio di codice mostra come utilizzare `import-playback-key-pair`.

### AWS CLI

Per importare la parte pubblica di una nuova coppia di chiavi

L'`import-playback-key-pair` esempio seguente importa la chiave pubblica specificata (specificata come stringa nel PEM formato) e restituisce l'arn e l'impronta digitale della nuova coppia di chiavi.

```
aws ivs import-playback-key-pair \
  --name "my-playback-key" \
  --public-key-material "G11bnQx0TA3BgNVBAMMFdoeSBhcmUgew91IGR1..."
```

Output:

```
{
  "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": {}
  }
}
```

```
}
```

Per maggiori informazioni, consulta [Impostazione dei canali privati](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, vedere [ImportPlaybackKeyPair](#) in AWS CLI Command Reference.

## list-channels

Il seguente esempio di codice mostra come utilizzare `list-channels`.

### AWS CLI

Esempio 1: Per ottenere informazioni di riepilogo su tutti i canali

L'`list-channels` seguente elenca tutti i canali del tuo AWS account.

```
aws ivs list-channels
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Creare un canale nella Guida](#) per l'utente IVS a bassa latenza.

Esempio 2: per ottenere informazioni di riepilogo su tutti i canali, filtrate in base al valore specificato RecordingConfiguration ARN

L'`list-channels` seguente elenca tutti i canali AWS del tuo account associati a quanto specificato RecordingConfiguration ARN.

```

aws ivs list-channels \
  --filter-by-recording-configuration-arn "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh"

```

Output:

```

{
  "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"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Record to Amazon S3](#) nella IVSLow-Latency User Guide.

Esempio 3: per ottenere informazioni di riepilogo su tutti i canali, filtrate in base a quanto specificato PlaybackRestrictionPolicy ARN

L'`list-channels` seguente elenca tutti i canali AWS del tuo account associati a quanto specificato PlaybackRestrictionPolicy ARN.

```
aws ivs list-channels \
  --filter-by-playback-restriction-policy-arn "arn:aws:ivs:us-
  west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ"
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Contenuto e visualizzatori indesiderati](#) nella Guida per l'utente IVSa bassa latenza.

- Per i API dettagli, vedere [ListChannels](#) in Command Reference.AWS CLI

## list-playback-key-pairs

Il seguente esempio di codice mostra come utilizzare `list-playback-key-pairs`.

AWS CLI

Per ottenere informazioni di riepilogo su tutte le coppie di tasti di riproduzione

L'`list-playback-key-pairs` seguente restituisce informazioni su tutte le coppie di chiavi.

```
aws ivs list-playback-key-pairs
```

Output:

```
{
  "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": {}
    }
  ]
}
```

Per maggiori informazioni, consulta [Impostazione dei canali privati](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, vedere [ListPlaybackKeyPairs](#) in AWS CLI Command Reference.

## `list-playback-restriction-policies`

Il seguente esempio di codice mostra come utilizzare `list-playback-restriction-policies`.

AWS CLI

Per ottenere informazioni di riepilogo su tutte le politiche di restrizione della riproduzione

L'`list-playback-restriction-policies` seguente elenca tutte le politiche di restrizione della riproduzione per il tuo account. AWS

```
aws ivs list-playback-restriction-policies
```

Output:



```
{
  "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"
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta [Contenuti e visualizzatori indesiderati nella Guida per l'utente a bassa latenza](#).

- Per i API dettagli, vedere [ListPlaybackRestrictionPolicies](#) in Command Reference.AWS CLI

## list-recording-configurations

Il seguente esempio di codice mostra come utilizzare `list-recording-configurations`.

### AWS CLI

Per elencare tutte le RecordingConfiguration risorse create in questo account

L'`list-recording-configurations` esempio seguente ottiene informazioni su tutte le RecordingConfiguration risorse del tuo account.

```
aws ivs list-recording-configurations
```

Output:

```
{
  "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": {}
    }
  ]
}
```

Per ulteriori informazioni, consulta [Record to Amazon S3](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [ListRecordingConfigurations AWS CLI](#) Command Reference.

## list-stream-keys

Il seguente esempio di codice mostra come utilizzare `list-stream-keys`.

### AWS CLI

Per ottenere un elenco di chiavi di streaming

L'`list-stream-keys` seguente elenca tutte le chiavi di stream per uno specifico ARN (Amazon Resource Name).

```
aws ivs list-stream-keys \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Output:

```
{  
  "streamKeys": [  
    {  
      "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",  
      "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",  
      "tags": {}  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Create a Channel](#) nella IVS Low-Latency User Guide.

- Per API i dettagli, vedere [ListStreamKeys](#) in AWS CLI Command Reference.

## `list-stream-sessions`

Il seguente esempio di codice mostra come utilizzare `list-stream-sessions`.

### AWS CLI

Per ottenere un riepilogo degli stream correnti e precedenti per un canale specificato nella regione corrente AWS

L'`list-stream-sessions` seguente riporta informazioni di riepilogo per gli stream per un canale specifico ARN (Amazon Resource Name).

```
aws ivs list-stream-sessions \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --max-results 25 \  
  --next-token ""
```

Output:

```
{
```

```
"nextToken": "set-2",
"streamSessions": [
  {
    "startTime": 1641578182,
    "endTime": 1641579982,
    "hasErrorEvent": false,
    "streamId": "mystream"
  }
  ...
]
}
```

Per ulteriori informazioni, consulta [Create a Channel](#) nella IVSLow-Latency User Guide.

- Per API i dettagli, vedere [ListStreamSessions](#) in AWS CLI Command Reference.

## list-streams

Il seguente esempio di codice mostra come utilizzare `list-streams`.

### AWS CLI

Per ottenere un elenco dei live streaming e del loro stato

L'`list-streams` esempio seguente elenca tutti i live streaming del tuo AWS account.

```
aws ivs list-streams
```

Output:

```
{
  "streams": [
    {
      "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
      "state": "LIVE",
      "health": "HEALTHY",
      "streamId": "st-ABCDEFghij01234KLMN5678",
      "viewerCount": 1
    }
  ]
}
```

Per ulteriori informazioni, consulta [Creare un canale nella Guida](#) per l'utente IVSa bassa latenza.

- Per API i dettagli, vedere [ListStreams](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare tutti i tag di una AWS risorsa (ad esempio: canale, chiave stream)

L'`list-tags-for-resource` esempio seguente elenca tutti i tag per una risorsa specificata ARN (Amazon Resource Name).

```
aws ivs list-tags-for-resource \
  --resource-arn arn:aws:ivs:us-west-2:12345689012:channel/abcdABCDefgh
```

Output:

```
{
  "tags":
  {
    "key1": "value1",
    "key2": "value2"
  }
}
```

Per ulteriori informazioni, consulta [Tagging](#) in Amazon Interactive Video Service API Reference.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI](#) Command Reference.

## put-metadata

Il seguente esempio di codice mostra come utilizzare `put-metadata`.

### AWS CLI

Per inserire metadati nello stream attivo per un canale specifico

L'`put-metadata` esempio seguente inserisce i metadati forniti nello stream per il canale specificato.

```
aws ivs put-metadata \
```

```
--channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
--metadata '{"my": "metadata"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Creare un canale](#) nella Guida per l'utente IVSa bassa latenza.

- Per API i dettagli, vedere [PutMetadata](#) in AWS CLI Command Reference.

## start-viewer-session-revocation

Il seguente esempio di codice mostra come utilizzare `start-viewer-session-revocation`.

### AWS CLI

Per revocare una sessione di visualizzazione per una determinata coppia di ID ARN e canali multipli

L'`start-viewer-session-revocation` esempio seguente avvia il processo di revoca della sessione di visualizzazione associata a un canale ARN e a un ID di visualizzatore specificati, fino all'inclusione del numero di versione della sessione specificato. Se la versione non viene fornita, il valore predefinito è 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
```

Questo comando non produce alcun output.

Per maggiori informazioni, consulta [Impostazione dei canali privati](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, vedere [StartViewerSessionRevocation](#) in AWS CLI Command Reference.

## stop-stream

Il seguente esempio di codice mostra come utilizzare `stop-stream`.

### AWS CLI

Per interrompere uno stream specificato

L'`stop-stream` seguente interrompe lo streaming sul canale specificato.

```
aws ivs stop-stream \  
  --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Creare un canale nella Guida](#) per l'utente IVS a bassa latenza.

- Per API i dettagli, vedere [StopStream](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere o aggiornare tag per una AWS risorsa (ad esempio: canale, chiave stream)

L'`tag-resource` seguente aggiunge o aggiorna i tag per una risorsa specificata ARN (Amazon Resource Name).

```
aws ivs tag-resource \  
  --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --tags "tagkey1=tagvalue1, tagkey2=tagvalue2"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging](#) in Amazon Interactive Video Service API Reference.

- Per API i dettagli, consulta [TagResource AWS CLI](#) Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una AWS risorsa (ad esempio: canale, chiave stream)

L'`untag-resource` seguente rimuove i tag specificati per una risorsa specificata ARN (Amazon Resource Name).

```
aws ivs untag-resource \  
  --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --tag-keys "tagkey1, tagkey2"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging](#) in Amazon Interactive Video Service API Reference.

- Per API i dettagli, consulta [UntagResource AWS CLI Command Reference](#).

## update-channel

Il seguente esempio di codice mostra come utilizzare `update-channel`.

### AWS CLI

Esempio 1: aggiornare le informazioni di configurazione di un canale

L'esempio seguente aggiorna la configurazione del canale per un canale specificato ARN per modificare il nome del canale. Ciò non influisce sullo streaming continuo di questo canale; è necessario interrompere e riavviare lo streaming affinché le modifiche abbiano effetto.

```
aws ivs update-channel \  
  --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \  
  --name "channel-1" \  
  --insecure-ingest
```

Output:

```
{  
  "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":  
        "AB1C2defGHijklMNop3PqQRstUvwxyzABCDefghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"    }  
  }  
}
```



```

    },
    "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": {}
}

```

Per ulteriori informazioni, consulta [Creare un canale nella Guida](#) per l'utente IVS a bassa latenza.

Esempio 2: aggiornare la configurazione di un canale per abilitare la registrazione

L'update-channel esempio seguente aggiorna la configurazione dei canali per un canale specifico ARN per abilitare la registrazione. Ciò non influisce sullo streaming continuo di questo canale; è necessario interrompere e riavviare lo streaming affinché le modifiche abbiano effetto.

```

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"

```

Output:

```

{
  "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":
"BA1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "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": {}
  }
}

```

Per ulteriori informazioni, consulta [Record to Amazon S3](#) nella IVSLow-Latency User Guide.

Esempio 3: aggiornare la configurazione di un canale per disabilitare la registrazione

L'update-channel esempio seguente aggiorna la configurazione dei canali per un canale specifico per ARN disabilitare la registrazione. Ciò non influisce sullo streaming continuo di questo canale; è necessario interrompere e riavviare lo streaming affinché le modifiche abbiano effetto.

```

aws ivs update-channel \
  --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \
  --recording-configuration-arn ""

```

Output:

```

{
  "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":
"AB1C2edfGHijklMN03PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "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": {}
  }
}

```

```
}
}
```

Per ulteriori informazioni, consulta [Record to Amazon S3](#) nella IVSLow-Latency User Guide.

Esempio 4: aggiornare la configurazione di un canale per abilitare la restrizione della riproduzione

L'update-channel esempio seguente aggiorna la configurazione dei canali per un canale specifico per ARN applicare una politica di restrizione della riproduzione. Ciò non influisce sullo streaming continuo di questo canale; è necessario interrompere e riavviare lo streaming affinché le modifiche abbiano effetto.

```
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"
```

Output:

```
{
  "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":
"AB1C2defGHijkLMNo3PqQRstUvwxyzaCBDEfghh4ijkLMN5opqrStuVWxyzAbCDEfghIJ"
    },
    "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": {}
  }
}
```

```
}
```

Per ulteriori informazioni, consulta [Contenuti e visualizzatori indesiderati](#) nella Guida per l'utente IVSa bassa latenza.

Esempio 5: aggiornare la configurazione di un canale per disabilitare la restrizione alla riproduzione

L'update-channel esempio seguente aggiorna la configurazione dei canali per un canale specifico per disabilitare la ARN restrizione alla riproduzione. Ciò non influisce sullo streaming continuo di questo canale; è necessario interrompere e riavviare lo streaming affinché le modifiche abbiano effetto.

```
aws ivs update-channel \  
  --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \  
  --playback-restriction-policy-arn ""
```

Output:

```
{  
  "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": {}  
  }  
}
```

Per ulteriori informazioni, consulta [Contenuti e visualizzatori indesiderati](#) nella Guida per l'utente IVSa bassa latenza.

- Per i API dettagli, vedere [UpdateChannel](#) in Command Reference.AWS CLI

## update-playback-restriction-policy

Il seguente esempio di codice mostra come utilizzare `update-playback-restriction-policy`.

### AWS CLI

Per aggiornare una politica di restrizione della riproduzione

L'`update-playback-restriction-policy`esempio seguente aggiorna la politica di restrizione della riproduzione con la politica specificata ARN per disabilitare l'applicazione rigorosa dell'origine. Ciò non influisce sullo streaming continuo del canale associato; è necessario interrompere e riavviare lo streaming affinché le modifiche abbiano effetto.

```
aws ivs update-playback-restriction-policy \
  --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
  ABCdef34ghIJ" \
  --no-enable-strict-origin-enforcement
```

Output:

```
{
  "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"
    }
  }
}
```

```
}  
  }  
}
```

Per ulteriori informazioni, consulta [Contenuti e visualizzatori indesiderati](#) nella Guida per l'utente IVSa bassa latenza.

- Per i API dettagli, vedere [UpdatePlaybackRestrictionPolicy](#) in Command Reference.AWS CLI

## Esempi di Amazon IVS Chat utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon IVS Chat. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-chat-token**

Il seguente esempio di codice mostra come utilizzare `create-chat-token`.

AWS CLI

Per creare un token di chat

L'`create-chat-token`esempio seguente crea un token di chat crittografato che viene utilizzato per stabilire una WebSocket connessione individuale a una stanza. Il token è valido per un minuto e una connessione (sessione) stabilita con il token è valida per la durata specificata.

```
aws ivschat create-chat-token \  
  --roomIdIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6", \  
  --durationSeconds 60
```

```
--userId "11231234" \  
--capabilities "SEND_MESSAGE", \  
--sessionDurationInMinutes 30
```

Output:

```
{  
  "token": "ACEGmnoq#1rstu2...BDFH3vxwy!4hlm!#5",  
  "sessionExpirationTime": "2022-03-16T04:44:09+00:00"  
  "state": "CREATING",  
  "tokenExpirationTime": "2022-03-16T03:45:09+00:00"  
}
```

Per ulteriori informazioni, consulta la [Fase 3: Autenticazione e autorizzazione dei client di chat nella Guida](#) per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta AWS CLI Command [CreateChatToken](#)Reference.

## create-logging-configuration

Il seguente esempio di codice mostra come utilizzare `create-logging-configuration`.

AWS CLI

Per creare una `LoggingConfiguration` risorsa di chat

L'esempio seguente crea una `LoggingConfiguration` risorsa che consente ai client di archiviare e registrare i messaggi inviati.

```
aws ivschat create-logging-configuration \  
  --destination-configuration s3={bucketName=demo-logging-bucket} \  
  --name "test-logging-config" \  
  --tags "key1=value1, key2=value2"
```

Output:

```
{  
  "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"
}

```

Per ulteriori informazioni, consulta la sezione Guida [introduttiva ad Amazon IVS Chat](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [CreateLoggingConfiguration AWS CLI Command Reference](#).

## create-room

Il seguente esempio di codice mostra come utilizzare `create-room`.

### AWS CLI

Per creare una stanza

L'`create-room`esempio seguente crea una nuova stanza.

```

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

```

Output:

```

{
  "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"  
}
```

Per ulteriori informazioni, consulta la [Fase 2: Creare una chat room](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [CreateRoom AWS CLI](#) Command Reference.

## delete-logging-configuration

Il seguente esempio di codice mostra come utilizzare `delete-logging-configuration`.

### AWS CLI

Per eliminare una `LoggingConfiguration` risorsa di chat

L'`delete-logging-configuration` esempio seguente elimina la `LoggingConfiguration` risorsa per quanto specificato ARN.

```
aws ivschat delete-logging-configuration \  
  --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
  ABcdef34ghIJ"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione Guida [introduttiva ad Amazon IVS Chat](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [DeleteLoggingConfiguration AWS CLI](#) Command Reference.

## delete-message

Il seguente esempio di codice mostra come utilizzare `delete-message`.

### AWS CLI

Per eliminare messaggi da una stanza specificata

L'`delete-message` esempio seguente invia un evento alla stanza specificata, che ordina ai client di eliminare il messaggio specificato, ovvero di annullarne la visualizzazione ed eliminarlo dalla cronologia chat del client.

```
aws ivschat delete-message \  
  --roomIdIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
  --id "ABC123def456" \  
  --reason "Message contains profanity"
```

Output:

```
{  
  "id": "12345689012"  
}
```

Per ulteriori informazioni, consulta la sezione Guida [introduttiva ad Amazon IVS Chat](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [DeleteMessage AWS CLI](#) Command Reference.

## delete-room

Il seguente esempio di codice mostra come utilizzare `delete-room`.

### AWS CLI

Per eliminare una stanza

L'esempio seguente elimina la stanza specificata. I client connessi vengono disconnessi. In caso di successo, restituisce HTTP 204 con un corpo di risposta vuoto.

```
aws ivschat delete-room \  
  --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione Guida [introduttiva ad Amazon IVS Chat](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [DeleteRoom AWS CLI](#) Command Reference.

## disconnect-user

Il seguente esempio di codice mostra come utilizzare `disconnect-user`.

## AWS CLI

Per disconnettere un utente da una stanza

L'`disconnect-user` seguente disconnette tutte le connessioni dell'utente specificato dalla stanza specificata. In caso di successo, restituisce HTTP 200 con un corpo di risposta vuoto.

```
aws ivschat disconnect-user \  
  --roomIdIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \  
  --userId "ABC123def456" \  
  --reason "Violated terms of service"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione Guida [introduttiva ad Amazon IVS Chat](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [DisconnectUser AWS CLI Command Reference](#).

## get-logging-configuration

Il seguente esempio di codice mostra come utilizzare `get-logging-configuration`.

## AWS CLI

Per ottenere informazioni su una `LoggingConfiguration` risorsa

L'`get-logging-configuration` seguente ottiene informazioni sulla `LoggingConfiguration` risorsa per l'oggetto specificato ARN.

```
aws ivschat get-logging-configuration \  
  --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
  ABcdef34ghIJ"
```

Output:

```
{  
  "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"
  }
```

Per ulteriori informazioni, consulta la sezione Guida [introduttiva ad Amazon IVS Chat](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [GetLoggingConfiguration AWS CLI](#) Command Reference.

## get-room

Il seguente esempio di codice mostra come utilizzare `get-room`.

### AWS CLI

Per ottenere la stanza specificata

L'`get-room` esempio seguente ottiene informazioni sulla stanza specificata.

```
aws ivschat get-room \
  --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"
```

Output:

```
{
  "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"
```

```
}
```

Per ulteriori informazioni, consulta la sezione Guida [introduttiva ad Amazon IVS Chat](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [GetRoom AWS CLI Command Reference](#).

## list-logging-configurations

Il seguente esempio di codice mostra come utilizzare `list-logging-configurations`.

### AWS CLI

Per ottenere informazioni di riepilogo su tutte le configurazioni di registrazione per l'utente nella AWS regione in cui viene elaborata la API richiesta

L'`list-logging-configurations` seguente elenca le informazioni su tutte le LoggingConfiguration risorse per l'utente nella AWS regione in cui viene elaborata la API richiesta.

```
aws ivschat list-logging-configurations \
  --max-results 2 \
  --next-token ""
```

### Output:

```
{
  "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"
      }
      ...
    ]
  }
```

Per ulteriori informazioni, consulta la sezione Guida [introduttiva ad Amazon IVS Chat](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [ListLoggingConfigurations AWS CLI](#) Command Reference.

## list-rooms

Il seguente esempio di codice mostra come utilizzare `list-rooms`.

### AWS CLI

Per ottenere informazioni di riepilogo su tutte le camere nella regione corrente

L'`list-rooms` esempio seguente ottiene informazioni di riepilogo su tutte le sale della AWS regione in cui viene elaborata la richiesta. I risultati vengono ordinati in ordine decrescente di `updateTime`

```
aws ivschat list-rooms \
  --logging-configuration-identifier "arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ" \
  --max-results 10 \
  --next-token ""
```

Output:

```
{
  "nextToken": "page3",
  "rooms": [
    {
      "arn:aws:ivschat:us-west-2:123456789012: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"
      }
    ]
  }
```

Per ulteriori informazioni, consulta la sezione Guida [introduttiva ad Amazon IVS Chat](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [ListRooms AWS CLI Command Reference](#).

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare tutti i tag di una AWS risorsa (ad esempio: Room)

L'`list-tags-for-resource` esempio seguente elenca tutti i tag per una risorsa specificata ARN (Amazon Resource Name).

```
aws ivschat list-tags-for-resource \
  --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6
```

Output:

```
{
  "tags":
  {
    "key1": "value1",
    "key2": "value2"
  }
}
```

Per ulteriori informazioni, consulta [Tagging](#) in Amazon Interactive Video Service API Reference.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## send-event

Il seguente esempio di codice mostra come utilizzare `send-event`.

## AWS CLI

Per inviare un evento a una stanza

L'`send-event` seguente invia l'evento specificato alla stanza specificata.

```
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."
```

Output:

```
{  
  "id": "12345689012"  
}
```

Per ulteriori informazioni, consulta la sezione Guida [introduttiva ad Amazon IVS Chat](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [SendEvent AWS CLI Command Reference](#).

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere o aggiornare i tag per una AWS risorsa (ad esempio: Room)

L'`tag-resource` seguente aggiunge o aggiorna i tag per una risorsa specificata ARN (Amazon Resource Name). In caso di successo, restituisce HTTP 200 con un corpo di risposta vuoto.

```
aws ivschat tag-resource \  
  --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \  
  --tags "tagkey1=tagkeyvalue1, tagkey2=tagkeyvalue2"
```

Questo comando non produce alcun output.



Per ulteriori informazioni, consulta [Tagging](#) in Amazon Interactive Video Service API Reference.

- Per API i dettagli, consulta [TagResource AWS CLI](#) Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una AWS risorsa (ad esempio: Room)

L'`untag-resource` esempio seguente rimuove i tag specificati per una risorsa specificata ARN (Amazon Resource Name). In caso di successo, restituisce HTTP 200 con un corpo di risposta vuoto.

```
aws ivschat untag-resource \  
  --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \  
  --tag-keys "tagkey1, tagkey2"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging](#) in Amazon Interactive Video Service API Reference.

- Per API i dettagli, consulta [UntagResource AWS CLI](#) Command Reference.

## update-logging-configuration

Il seguente esempio di codice mostra come utilizzare `update-logging-configuration`.

### AWS CLI

Per aggiornare la configurazione di registrazione di una stanza

L'`update-logging-configuration` esempio seguente aggiorna una `LoggingConfiguration` risorsa con i dati forniti.

```
aws ivschat update-logging-configuration \  
  --destination-configuration s3={bucketName=demo-logging-bucket} \  
  --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/  
ABcdef34ghIJ" \  
  --name "test-logging-config"
```

## Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta la sezione Guida [introduttiva ad Amazon IVS Chat](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [UpdateLoggingConfiguration AWS CLI Command Reference](#).

## update-room

Il seguente esempio di codice mostra come utilizzare `update-room`.

### AWS CLI

Per aggiornare la configurazione di una stanza

L'esempio seguente aggiorna la configurazione della stanza specificata con i dati forniti.

```
aws ivschat update-room \
  --identifier "arn:aws:ivschat:us-west-2:123456789012: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
```

## Output:

```
{
  "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-2:12345689012:logging-configuration/ABcdef34ghIJ"],
  "maximumMessageLength": 256,
  "maximumMessageRatePerSecond": 5,
  "name": "chat-room-a",
  "tags": {},
  "updateTime": "2022-03-16T07:22:09+00:00"
}
```

Per ulteriori informazioni, consulta la sezione Guida [introduttiva ad Amazon IVS Chat](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [UpdateRoom AWS CLI](#) Command Reference.

## Esempi di Amazon IVS Real-Time Streaming utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon IVS Real-Time Streaming. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **create-encoder-configuration**

Il seguente esempio di codice mostra come utilizzare `create-encoder-configuration`.

## AWS CLI

Per creare una configurazione del codificatore di composizione

L'`create-encoder-configuration` seguente crea una configurazione del codificatore di composizione con le proprietà specificate.

```
aws ivs-realtime create-encoder-configuration \  
  --name test-ec --video bitrate=3500000,framerate=30.0,height=1080,width=1920
```

Output:

```
{  
  "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  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [CreateEncoderConfiguration AWS CLI Command Reference](#).

## **create-participant-token**

Il seguente esempio di codice mostra come utilizzare `create-participant-token`.

## AWS CLI

Per creare un token per i partecipanti allo stage

L'`create-participant-token` seguente crea un token di partecipazione per la fase specificata.

```
aws ivs-realtime create-participant-token \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --user-id bob
```

Output:

```
{  
  "participantToken": {  
    "expirationTime": "2023-03-07T09:47:43+00:00",  
    "participantId": "ABCDEFghij01234KLMN6789",  
    "token": "abcd1234defg5678"  
  }  
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [CreateParticipantToken AWS CLI Command Reference](#).

## create-stage

Il seguente esempio di codice mostra come utilizzare `create-stage`.

### AWS CLI

Esempio 1: creare uno stage

L'`create-stage` seguente crea uno stage e un token per i partecipanti allo stage per un utente specificato.

```
aws ivs-realtime create-stage \  
  --name stage1 \  
  --participant-token-configurations userId=alice
```

Output:

```
{  
  "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": {}
    }
  }
}

```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

Esempio 2: creare uno stage e configurare la registrazione dei singoli partecipanti

L'create-stageesempio seguente crea uno stage e configura la registrazione dei singoli partecipanti.

```

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"}'

```

Output:

```

{
  "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": {}
  }
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [CreateStage AWS CLI Command Reference](#).

## create-storage-configuration

Il seguente esempio di codice mostra come utilizzare `create-storage-configuration`.

### AWS CLI

Per creare una configurazione di archiviazione della composizione

L'esempio seguente crea una configurazione di archiviazione della composizione con le proprietà specificate.

```
aws ivs-realtime create-storage-configuration \
  --name "test-sc" --s3 "bucketName=test-bucket-name"
```

Output:

```
{
  "storageConfiguration": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/ABabCDcdEFef",
    "name": "test-sc",
    "s3": {
      "bucketName": "test-bucket-name"
    },
    "tags": {}
  }
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [CreateStorageConfiguration AWS CLI](#) Command Reference.

## delete-encoder-configuration

Il seguente esempio di codice mostra come utilizzare `delete-encoder-configuration`.

### AWS CLI

Per eliminare una configurazione del codificatore di composizione

Quanto segue `delete-encoder-configuration` elimina la configurazione del codificatore di composizione specificata dal dato (ARN Amazon Resource Name).

```
aws ivs-realtime delete-encoder-configuration \  
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/  
  ABabCDcdEFef"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [DeleteEncoderConfiguration AWS CLI](#) Command Reference.

## delete-public-key

Il seguente esempio di codice mostra come utilizzare `delete-public-key`.

### AWS CLI

Per eliminare una chiave pubblica

Quanto segue `delete-public-key` elimina la chiave pubblica specificata.

```
aws ivs-realtime delete-public-key \  
  --arn arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2
```

Questo comando non produce alcun output.



Per ulteriori informazioni, consulta [Distribuire i token dei partecipanti](#) nella Guida per l'utente di Amazon IVS Real-Time Streaming.

- Per API i dettagli, consulta AWS CLI Command [DeletePublicKey](#) Reference.

## delete-stage

Il seguente esempio di codice mostra come utilizzare `delete-stage`.

### AWS CLI

Per eliminare una fase

L'`delete-stage` esempio seguente elimina lo stadio specificato.

```
aws ivs-realtime delete-stage \  
  --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [DeleteStage AWS CLI](#) Command Reference.

## delete-storage-configuration

Il seguente esempio di codice mostra come utilizzare `delete-storage-configuration`.

### AWS CLI

Per eliminare una configurazione di memorizzazione della composizione

Quanto segue `delete-storage-configuration` elimina la configurazione di archiviazione della composizione specificata dal dato ARN (Amazon Resource Name).

```
aws ivs-realtime delete-storage-configuration \  
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/  
  ABabCDcdEFef"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [DeleteStorageConfiguration AWS CLI](#) Command Reference.

## **disconnect-participant**

Il seguente esempio di codice mostra come utilizzare `disconnect-participant`.

### AWS CLI

Per disconnettere un partecipante al palco

L'`disconnect-participant` esempio seguente disconnette il partecipante specificato dalla fase specificata.

```
aws ivs-realtime disconnect-participant \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --participant-id ABCDEFghij01234KLMN5678
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [DisconnectParticipant AWS CLI](#) Command Reference.

## **get-composition**

Il seguente esempio di codice mostra come utilizzare `get-composition`.

### AWS CLI

Esempio 1: Per ottenere una composizione con le impostazioni di layout predefinite

L'`get-composition` esempio seguente ottiene la composizione per ARN (Amazon Resource Name) specificato.

```
aws ivs-realtime get-composition \  
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

## Output:

```

{
  "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/GHFabcgfABC/"
          }
        },
        "id": "GHFabcgfABC",
        "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": {}
}
}

```

Per ulteriori informazioni, consulta [Composite Recording \(Real-Time Streaming\)](#) nella Guida per l'utente di Amazon Interactive Video Service.

Esempio 2: per ottenere una composizione con layout PiP

L'get-compositionesempio seguente ottiene la composizione per il ARN (Amazon Resource Name) specificato, che utilizza il layout PiP.

```

aws ivs-realtime get-composition \
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs"

```

Output:

```

{
  "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": {}
  }
}
```

Per ulteriori informazioni, consulta [Composite Recording \(Real-Time Streaming\)](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [GetComposition AWS CLI Command Reference](#).

## get-encoder-configuration

Il seguente esempio di codice mostra come utilizzare `get-encoder-configuration`.

### AWS CLI

Per ottenere una configurazione del codificatore di composizione

L'`get-encoder-configuration` esempio seguente ottiene la configurazione del codificatore di composizione specificata dal dato ARN (Amazon Resource Name).

```
aws ivs-realtime get-encoder-configuration \
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh"
```

Output:

```
{
  "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
    }
  }
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [GetEncoderConfiguration AWS CLI Command Reference](#).

## get-participant

Il seguente esempio di codice mostra come utilizzare `get-participant`.

### AWS CLI

Per ottenere un partecipante al palco

L'`get-participant` esempio seguente ottiene il partecipante alla fase per un ID partecipante e un ID sessione specificati nella fase specificata (ARN Amazon Resource Name).

```
aws ivs-realtime get-participant \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --session-id st-a1b2c3d4e5f6g \  
  --participant-id abCDEf12GHIj
```

Output:

```
{  
  "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": "",  
  }  
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [GetParticipant AWS CLI Command Reference](#).

## get-public-key

Il seguente esempio di codice mostra come utilizzare `get-public-key`.

### AWS CLI

Per utilizzare una chiave pubblica esistente per firmare i token dei partecipanti allo stage

L'`get-public-key` seguente ottiene una chiave pubblica specificata dal provider ARN, per firmare i token dei partecipanti alla fase.

```
aws ivs-realtime get-public-key \  
  --arn arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2
```

Output:

```
{  
  "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": {}  
  }  
}
```

Per ulteriori informazioni, consulta [Distribuire i token dei partecipanti](#) nella Guida per l'utente di Amazon IVS Real-Time Streaming.

- Per API i dettagli, consulta AWS CLI Command [GetPublicKey](#) Reference.

## get-stage-session

Il seguente esempio di codice mostra come utilizzare `get-stage-session`.



## AWS CLI

Per partecipare a una sessione sul palco

L'`get-stage-session` seguente ottiene la sessione di stage per un ID di sessione specificato di una fase specificata ARN (Amazon Resource Name).

```
aws ivs-realtime get-stage-session \  
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \  
  --session-id st-a1b2c3d4e5f6g
```

Output:

```
{  
  "stageSession": {  
    "endTime": "2023-04-26T20:36:29+00:00",  
    "sessionId": "st-a1b2c3d4e5f6g",  
    "startTime": "2023-04-26T20:30:29.602000+00:00"  
  }  
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [GetStageSession AWS CLI](#) Command Reference.

## get-stage

Il seguente esempio di codice mostra come utilizzare `get-stage`.

## AWS CLI

Per ottenere le informazioni di configurazione di una fase

L'`get-stage` seguente ottiene la configurazione della fase per una fase specificata ARN (Amazon Resource Name).

```
aws ivs-realtime get-stage \  
  --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

Output:

```
{
  "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": {}
  }
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [GetStage AWS CLI Command Reference](#).

## get-storage-configuration

Il seguente esempio di codice mostra come utilizzare `get-storage-configuration`.

### AWS CLI

Per ottenere una configurazione di archiviazione della composizione

L'esempio seguente ottiene la configurazione di archiviazione della composizione specificata dal dato ARN (Amazon Resource Name).

```
aws ivs-realtime get-storage-configuration \
  --name arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/abcdABCDefgh"
```

Output:

```
{
  "storageConfiguration": {
    "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
abcdABCDefgh",
    "name": "test-sc",
    "s3": {
      "bucketName": "test-bucket-name"
    },
    "tags": {}
  }
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [GetStorageConfiguration AWS CLI Command Reference](#).

## import-public-key

Il seguente esempio di codice mostra come utilizzare `import-public-key`.

### AWS CLI

Per importare una chiave pubblica esistente da utilizzare per firmare i token dei partecipanti allo stage

L'esempio seguente importa una chiave pubblica da un file di materiale, da utilizzare per firmare i token dei partecipanti alla fase.

```
aws ivs-realtime import-public-key \
  --public-key-material="`cat public.pem`"
```

Output:

```
{
  "publicKey": {
    "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
    "name": "",
    "publicKeyMaterial": "-----BEGIN PUBLIC KEY-----
\nMHYwEAYHkoZIZj0CAQYFK4EEACIDYgAEqVWUtqs6EktQMR1sCYmEzGvRwtaycI16\n9pmzcpIWu/
uhNStGlteJ5odRfRwVkoQUMnSZXTCcbn9bBTTmiWo4mJcF00AzsthH
\n0UAb8NdD4tUE0At4a9hYP9IETEXAMPLE\n-----END PUBLIC KEY-----",
  }
}
```

```
    "fingerprint": "12:a3:44:56:bc:7d:e8:9f:10:2g:34:hi:56:78:90:12",
    "tags": {}
  }
}
```

Per ulteriori informazioni, consulta [Distribuire i token dei partecipanti](#) nella Guida per l'utente di Amazon IVS Real-Time Streaming.

- Per API i dettagli, consulta AWS CLI Command [ImportPublicKeyReference](#).

## list-compositions

Il seguente esempio di codice mostra come utilizzare `list-compositions`.

### AWS CLI

Per ottenere un elenco di composizioni

Di seguito `list-compositions` sono elencate tutte le composizioni per il tuo AWS account, nella AWS regione in cui viene elaborata la API richiesta.

```
aws ivs-realtime list-compositions
```

Output:

```
{
  "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": {}
    }
  ]
}

```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [ListCompositions AWS CLI Command Reference](#).

## list-encoder-configurations

Il seguente esempio di codice mostra come utilizzare `list-encoder-configurations`.

### AWS CLI

Per elencare le configurazioni del codificatore di composizione

Di seguito `list-encoder-configurations` sono elencate tutte le configurazioni del codificatore di composizione per il tuo AWS account, nella AWS regione in cui viene elaborata la API richiesta.

```
aws ivs-realtime list-encoder-configurations
```

Output:

```
{
  "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": {}
    }
  ]
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [ListEncoderConfigurations AWS CLI](#) Command Reference.

## list-participant-events

Il seguente esempio di codice mostra come utilizzare `list-participant-events`.

AWS CLI

Per ottenere un elenco degli eventi che hanno partecipato al palco

L'`list-participant-events` esempio seguente elenca tutti gli eventi dei partecipanti per un ID partecipante e un ID sessione specifici di una fase specifica (ARN Amazon Resource Name).

```
aws ivs-realtime list-participant-events \
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
  --session-id st-a1b2c3d4e5f6g \
  --participant-id abCDEf12GHIj
```

**Output:**

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [ListParticipantEvents AWS CLI Command Reference](#).

**list-participants**

Il seguente esempio di codice mostra come utilizzare `list-participants`.

**AWS CLI**

Per ottenere un elenco dei partecipanti allo stage

L'`list-participants` esempio seguente elenca tutti i partecipanti per un ID di sessione specificato di una fase specifica ARN (Amazon Resource Name).

```
aws ivs-realtime list-participants \
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
  --session-id st-a1b2c3d4e5f6g
```

Output:

```
{
  "participants": [
    {
      "firstJoinTime": "2023-04-26T20:30:34+00:00",
      "participantId": "abCDEf12GHIj"
      "published": true,
      "recordingState": "STOPPED",
      "state": "DISCONNECTED",
      "userId": ""
    }
  ]
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [ListParticipants AWS CLI](#) Command Reference.

## list-public-keys

Il seguente esempio di codice mostra come utilizzare `list-public-keys`.

AWS CLI

Per elencare le chiavi pubbliche esistenti disponibili per firmare i token dei partecipanti allo stage

L'`list-public-keys` esempio seguente elenca tutte le chiavi pubbliche disponibili per la firma dei token dei partecipanti alla fase, nella AWS regione in cui viene elaborata la richiesta. API

```
aws ivs-realtime list-public-keys
```

Output:

```
{
  "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": {}
    }
]
}

```

Per ulteriori informazioni, consulta [Distribuire i token dei partecipanti](#) nella Guida per l'utente di Amazon IVS Real-Time Streaming.

- Per API i dettagli, consulta AWS CLI Command [ListPublicKeys](#) Reference.

## list-stage-sessions

Il seguente esempio di codice mostra come utilizzare `list-stage-sessions`.

### AWS CLI

Per ottenere un elenco delle sessioni sul palco

L'`list-stage-session` esempio seguente elenca tutte le sessioni per una fase specificata ARN (Amazon Resource Name).

```

aws ivs-realtime list-stage-sessions \
  --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh

```

Output:

```

{
  "stageSessions": [
    {
      "endTime": "2023-04-26T20:36:29+00:00",
      "sessionId": "st-a1b2c3d4e5f6g",
      "startTime": "2023-04-26T20:30:29.602000+00:00"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [ListStageSessions AWS CLI Command Reference](#).

## list-stages

Il seguente esempio di codice mostra come utilizzare `list-stages`.

### AWS CLI

Per ottenere informazioni di riepilogo su tutte le fasi

L'`list-stages` esempio seguente elenca tutte le fasi del tuo AWS account, nella AWS regione in cui viene elaborata la API richiesta.

```
aws ivs-realtime list-stages
```

Output:

```
{
  "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": {}
    }
  ]
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [ListStages AWS CLI Command Reference](#).

## list-storage-configurations

Il seguente esempio di codice mostra come utilizzare `list-storage-configurations`.

### AWS CLI

Per elencare le configurazioni di archiviazione della composizione

Di seguito `list-storage-configurations` sono elencate tutte le configurazioni di archiviazione della composizione per il tuo AWS account, nella AWS regione in cui viene elaborata la API richiesta.

```
aws ivs-realtime list-storage-configurations
```

Output:

```
{
  "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": {}
    }
  ]
}
```

```
}

```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [ListStorageConfigurations AWS CLI](#) Command Reference.

## start-composition

Il seguente esempio di codice mostra come utilizzare `start-composition`.

### AWS CLI

Esempio 1: Per iniziare una composizione con le impostazioni di layout predefinite

L'`start-composition` esempio seguente avvia una composizione per lo stage specificato da trasmettere in streaming nelle posizioni specificate.

```
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"}}]'
```

Output:

```
{
  "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": {}
}

```

```
}

```

Per ulteriori informazioni, consulta [Composite Recording \(Real-Time Streaming\)](#) nella Guida per l'utente di Amazon Interactive Video Service.

Esempio 2: iniziare una composizione con il layout PiP

L'`start-composition` seguente avvia una composizione per lo stage specificato da trasmettere in streaming alle posizioni specificate utilizzando il layout PiP.

```
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}"'
```

Output:

```
{
  "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": {}
}
}

```

Per ulteriori informazioni, consulta [Composite Recording \(Real-Time Streaming\)](#) nella Guida per l'utente di Amazon Interactive Video Service.

- Per API i dettagli, consulta [StartComposition AWS CLI Command Reference](#).

## stop-composition

Il seguente esempio di codice mostra come utilizzare `stop-composition`.

### AWS CLI

Per interrompere una composizione

Quanto segue `stop-composition` interrompe la composizione specificata dal dato ARN (Amazon Resource Name).

```
aws ivs-realtime stop-composition \  
  --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [StopComposition AWS CLI Command Reference](#).

## update-stage

Il seguente esempio di codice mostra come utilizzare `update-stage`.

### AWS CLI

Per aggiornare la configurazione di una fase

L'update-stage esempio seguente aggiorna una fase per una fase specifica per ARN aggiornare il nome della fase e configurare la registrazione dei singoli partecipanti.

```
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
```

Output:

```
{
  "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": {}
  }
}
```

Per ulteriori informazioni, consulta [Enabling Multiple Hosts on an Amazon IVS Stream](#) nella Amazon Interactive Video Service User Guide.

- Per API i dettagli, consulta [UpdateStage AWS CLI](#) Command Reference.

## Esempi di utilizzo di Amazon Kendra AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Kendra. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### create-data-source

Il seguente esempio di codice mostra come utilizzare `create-data-source`.

#### AWS CLI

Per creare un connettore di origine dati Amazon Kendra

Quanto segue `create-data-source` crea e configura un connettore di origine dati Amazon Kendra. Puoi utilizzarlo `describe-data-source` per visualizzare lo stato di un connettore di origine dati e leggere eventuali messaggi di errore se lo stato mostra un connettore di origine dati "FAILED" da creare completamente.

```
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"]}'
```

Output:

```
{  
  "Id": "exampledatasource1"
```

```
}

```

Per ulteriori informazioni, consulta la sezione [Guida introduttiva all'utilizzo di un connettore per indici e sorgenti dati Amazon Kendra](#) nella Amazon Kendra Developer Guide.

- Per i API dettagli, consulta [CreateDataSource](#) Command Reference.AWS CLI

## create-index

Il seguente esempio di codice mostra come utilizzare `create-index`.

### AWS CLI

Per creare un indice Amazon Kendra

Quanto segue `create-index` crea e configura un indice Amazon Kendra. Puoi utilizzarlo `describe-index` per visualizzare lo stato di un indice e leggere eventuali messaggi di errore se lo stato mostra un indice "FAILED" da creare completamente.

```
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"}}'
```

Output:

```
{
  "Id": index1
}
```

Per ulteriori informazioni, consulta la sezione [Guida introduttiva all'utilizzo di un connettore per indici e sorgenti dati Amazon Kendra](#) nella Amazon Kendra Developer Guide.

- Per i API dettagli, consulta [CreateIndex](#) Command Reference.AWS CLI

## describe-data-source

Il seguente esempio di codice mostra come utilizzare `describe-data-source`.

### AWS CLI

Per ottenere informazioni su un connettore di origine dati Amazon Kendra

Di seguito `describe-data-source` vengono fornite informazioni su un connettore di origine dati Amazon Kendra. Puoi visualizzare la configurazione di un connettore di origine dati e leggere eventuali messaggi di errore se lo stato mostra un connettore di origine dati "FAILED" da creare completamente.

```
aws kendra describe-data-source \  
  --id exampledatasource1 \  
  --index-id exampleindex1
```

Output:

```
{  
  "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"]
}
}
```

Per ulteriori informazioni, consulta la sezione [Guida introduttiva all'utilizzo di un connettore per indici e sorgenti dati Amazon Kendra](#) nella Amazon Kendra Developer Guide.

- Per i API dettagli, consulta [DescribeDataSource](#) Command Reference.AWS CLI

## describe-index

Il seguente esempio di codice mostra come utilizzare `describe-index`.

### AWS CLI

Per ottenere informazioni su un indice Amazon Kendra

Di seguito `describe-index` vengono fornite informazioni su un indice Amazon Kendra. Puoi visualizzare la configurazione di un indice e leggere eventuali messaggi di errore se lo stato mostra un indice "FAILED" da creare completamente.

```
aws kendra describe-index \
  --id exampleindex1
```

Output:

```
{
  "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"
      }
    }
  ]
}

```

Per ulteriori informazioni, consulta la sezione [Guida introduttiva all'utilizzo di un connettore per indici e sorgenti dati Amazon Kendra](#) nella Amazon Kendra Developer Guide.

- Per i API dettagli, consulta [DescribeIndex](#) Command Reference.AWS CLI



## update-data-source

Il seguente esempio di codice mostra come utilizzare `update-data-source`.

### AWS CLI

Per aggiornare un connettore di origine dati Amazon Kendra

Quanto segue `update-data-source` aggiorna la configurazione di un connettore di origine dati Amazon Kendra. Se l'azione ha esito positivo, il servizio non restituisce alcun output, il codice di HTTP stato 200 o il codice AWS CLI restituito 0. È possibile utilizzare `describe-data-source` per visualizzare la configurazione e lo stato di un connettore di origine dati.

```
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"]}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Guida introduttiva all'utilizzo di un connettore per indici e sorgenti dati Amazon Kendra](#) nella Amazon Kendra Developer Guide.

- Per i API dettagli, consulta [UpdateDataSource](#) Command Reference.AWS CLI

## update-index

Il seguente esempio di codice mostra come utilizzare `update-index`.

## AWS CLI

Per aggiornare un indice Amazon Kendra

Quanto segue `update-index` aggiorna la configurazione di un indice Amazon Kendra. Se l'azione ha esito positivo, il servizio non restituisce alcun output, il codice di HTTP stato 200 o il codice AWS CLI restituito 0. È possibile utilizzare `describe-index` per visualizzare la configurazione e lo stato di un indice.

```
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"}}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Guida introduttiva all'utilizzo di un connettore per indici e sorgenti dati Amazon Kendra](#) nella Amazon Kendra Developer Guide.

- Per i API dettagli, consulta [UpdateIndexCommand Reference](#).AWS CLI

## Esempi di Kinesis che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface con Kinesis.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **add-tags-to-stream**

Il seguente esempio di codice mostra come utilizzare `add-tags-to-stream`.

#### AWS CLI

Per aggiungere tag a un flusso di dati

L'`add-tags-to-stream` esempio seguente assegna un tag con la chiave `samplekey` e il valore `example` allo stream specificato.

```
aws kinesis add-tags-to-stream \  
  --stream-name samplestream \  
  --tags samplekey=example
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Your Streams](#) nella Amazon Kinesis Data Streams Developer Guide.

- Per i API dettagli, consulta [AddTagsToStream](#) Command Reference.AWS CLI

### **create-stream**

Il seguente esempio di codice mostra come utilizzare `create-stream`.

#### AWS CLI

Per creare un flusso di dati

L'`create-stream` esempio seguente crea un flusso di dati denominato `samplestream` con 3 shard.

```
aws kinesis create-stream \  
  --stream-name samplestream \  
  --shards 3
```

```
--stream-name samplestream \  
--shard-count 3
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creating a Stream](#) nella Amazon Kinesis Data Streams Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateStreamReference](#).

## decrease-stream-retention-period

Il seguente esempio di codice mostra come utilizzare `decrease-stream-retention-period`.

### AWS CLI

Per ridurre il periodo di conservazione del flusso di dati

L'`decrease-stream-retention-period` esempio seguente riduce il periodo di conservazione (il periodo di tempo in cui i record di dati sono accessibili dopo essere stati aggiunti allo stream) di uno stream denominato `samplestream` a 48 ore.

```
aws kinesis decrease-stream-retention-period \  
--stream-name samplestream \  
--retention-period-hours 48
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Modifica del periodo di conservazione dei dati](#) nella Amazon Kinesis Data Streams Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DecreaseStreamRetentionPeriodReference](#).

## delete-stream

Il seguente esempio di codice mostra come utilizzare `delete-stream`.

### AWS CLI

Per eliminare un flusso di dati

L'`delete-stream` seguente elimina il flusso di dati specificato.

```
aws kinesis delete-stream \  
  --stream-name samplestream
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di uno stream](#) nella Amazon Kinesis Data Streams Developer Guide.

- Per API i dettagli, consulta Command [DeleteStream](#) Reference AWS CLI .

## **deregister-stream-consumer**

Il seguente esempio di codice mostra come utilizzare `deregister-stream-consumer`.

AWS CLI

Per annullare la registrazione di un consumatore di flussi di dati

L'`deregister-stream-consumer` seguente annulla la registrazione del consumatore specificato dal flusso di dati specificato.

```
aws kinesis deregister-stream-consumer \  
  --stream-arn arn:aws:kinesis:us-west-2:123456789012:stream/samplestream \  
  --consumer-name KinesisConsumerApplication
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Developing Consumer with Enhanced Fan-Out Using the Kinesis Data Streams nella Amazon Kinesis Data API](#) Streams Developer Guide.

- Per i API dettagli, consulta Command Reference. [DeregisterStreamConsumer](#) AWS CLI

## **describe-limits**

Il seguente esempio di codice mostra come utilizzare `describe-limits`.

AWS CLI

Per descrivere i limiti degli shard

L'`describe-limits` esempio seguente mostra i limiti e l'utilizzo degli shard per l'account corrente AWS .

```
aws kinesis describe-limits
```

Output:

```
{
  "ShardLimit": 500,
  "OpenShardCount": 29
}
```

Per ulteriori informazioni, consulta [Resharding a Stream nella Amazon Kinesis Data Streams Developer Guide](#).

- Per i API dettagli, consulta [DescribeLimits](#) Command Reference.AWS CLI

## `describe-stream-consumer`

Il seguente esempio di codice mostra come utilizzare `describe-stream-consumer`.

AWS CLI

Per descrivere un consumatore di flussi di dati

L'`describe-stream-consumer` esempio seguente restituisce la descrizione del consumatore specificato, registrato con il flusso di dati specificato.

```
aws kinesis describe-stream-consumer \
  --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \
  --consumer-name KinesisConsumerApplication
```

Output:

```
{
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta la sezione [Lettura dei dati da Amazon Kinesis Data Streams nella Amazon Kinesis Data Streams Developer Guide](#).

- Per i API dettagli, consulta [DescribeStreamConsumer](#) Command Reference.AWS CLI

## describe-stream-summary

Il seguente esempio di codice mostra come utilizzare `describe-stream-summary`.

### AWS CLI

Per descrivere un riepilogo del flusso di dati

L'`describe-stream-summary` esempio seguente fornisce una descrizione riepilogativa (senza l'elenco degli shard) del flusso di dati specificato.

```
aws kinesis describe-stream-summary \  
  --stream-name samplestream
```

Output:

```
{  
  "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  
  }  
}
```

```
}
```

Per ulteriori informazioni, consulta [Creating and Managing Streams](#) nella Amazon Kinesis Data Streams Developer Guide.

- Per API i dettagli, consulta Command [DescribeStreamSummary](#)Reference AWS CLI .

## describe-stream

Il seguente esempio di codice mostra come utilizzare `describe-stream`.

### AWS CLI

Per descrivere un flusso di dati

L'`describe-stream`esempio seguente restituisce i dettagli del flusso di dati specificato.

```
aws kinesis describe-stream \  
  --stream-name samplestream
```

Output:

```
{  
  "StreamDescription": {  
    "Shards": [  
      {  
        "ShardId": "shardId-000000000000",  
        "HashKeyRange": {  
          "StartingHashKey": "0",  
          "EndingHashKey": "113427455640312821154458202477256070484"  
        },  
        "SequenceNumberRange": {  
          "StartingSequenceNumber":  
"49600871682957036442365024926191073437251060580128653314"  
        }  
      },  
      {  
        "ShardId": "shardId-000000000001",  
        "HashKeyRange": {  
          "StartingHashKey": "113427455640312821154458202477256070485",  
          "EndingHashKey": "226854911280625642308916404954512140969"  
        },  
        "SequenceNumberRange": {
```



```

        "StartingSequenceNumber":
"4960087168297933718756355549332609155523708941634633746"
    }
  },
  {
    "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
}
}

```

Per ulteriori informazioni, consulta [Creating and Managing Streams](#) nella Amazon Kinesis Data Streams Developer Guide.

- Per API i dettagli, consulta Command [DescribeStream](#)Reference AWS CLI .

## disable-enhanced-monitoring

Il seguente esempio di codice mostra come utilizzare `disable-enhanced-monitoring`.

### AWS CLI

Per disabilitare il monitoraggio avanzato per le metriche a livello di shard

L'`disable-enhanced-monitoring` seguente disabilita il monitoraggio avanzato del flusso di dati Kinesis per le metriche a livello di shard.

```
aws kinesis disable-enhanced-monitoring \  
  --stream-name samplestream --shard-level-metrics ALL
```

Output:

```
{  
  "StreamName": "samplestream",  
  "CurrentShardLevelMetrics": [  
    "IncomingBytes",  
    "OutgoingRecords",  
    "IteratorAgeMilliseconds",  
    "IncomingRecords",  
    "ReadProvisionedThroughputExceeded",  
    "WriteProvisionedThroughputExceeded",  
    "OutgoingBytes"  
  ],  
  "DesiredShardLevelMetrics": []  
}
```

Per ulteriori informazioni, consulta [Monitoring Streams in Amazon Kinesis Data Streams nella Amazon Kinesis Data Streams Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [DisableEnhancedMonitoring](#) AWS CLI

## **enable-enhanced-monitoring**

Il seguente esempio di codice mostra come utilizzare `enable-enhanced-monitoring`.

AWS CLI

Per abilitare il monitoraggio avanzato delle metriche a livello di shard

L'`enable-enhanced-monitoring` seguente abilita il monitoraggio avanzato del flusso di dati Kinesis per le metriche a livello di shard.

```
aws kinesis enable-enhanced-monitoring \  
  --stream-name samplestream \  
  --shard-level-metrics ALL
```

**Output:**

```
{
  "StreamName": "samplestream",
  "CurrentShardLevelMetrics": [],
  "DesiredShardLevelMetrics": [
    "IncomingBytes",
    "OutgoingRecords",
    "IteratorAgeMilliseconds",
    "IncomingRecords",
    "ReadProvisionedThroughputExceeded",
    "WriteProvisionedThroughputExceeded",
    "OutgoingBytes"
  ]
}
```

Per ulteriori informazioni, consulta [Monitoring Streams in Amazon Kinesis Data Streams nella Amazon Kinesis Data Streams Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [EnableEnhancedMonitoring](#) AWS CLI

**get-records**

Il seguente esempio di codice mostra come utilizzare `get-records`.

**AWS CLI**

Per ottenere record da un frammento

L'`get-records` esempio seguente ottiene i record di dati dallo shard di un flusso di dati Kinesis utilizzando l'iteratore shard specificato.

```
aws kinesis get-records \
  --shard-iterator AAAAAAAAAAAF7/0mWD7IuHj1yGv/  

TKuNgx2ukD5xipCY4cy4gU96orWwZwcSXh3K9tAmGYe0ZyLZrvzze0FVf9iN99hUPw/w/  

b0YWYeefNvnf1DYt5XpDJghLKr3DzgzknTmMymDP3R+3wRKeuEw6/kdxY2yKJH0veaiekaVc4N2VwK/  

GvaGP2Hh9Fg7N++q0Adg6fIDQPt4p8RpavDbk+A4sL9SWG1
```

**Output:**

```
{
  "Records": [],
```

```
"MillisBehindLatest": 80742000
}
```

Per ulteriori informazioni, consulta [Developing Consumer Using the Kinesis API Data Streams AWS SDK with the](#) for Java nella Amazon Kinesis Data Streams Developer Guide.

- Per i API dettagli, consulta [GetRecords](#) Command Reference.AWS CLI

## get-shard-iterator

Il seguente esempio di codice mostra come utilizzare `get-shard-iterator`.

### AWS CLI

Per ottenere un iteratore shard

L'`get-shard-iterator` esempio seguente utilizza il tipo `AT_SEQUENCE_NUMBER` shard iterator e genera un iteratore shard per iniziare a leggere i record di dati esattamente dalla posizione indicata dal numero di sequenza specificato.

```
aws kinesis get-shard-iterator \
  --stream-name samplestream \
  --shard-id shardId-000000000001 \
  --shard-iterator-type LATEST
```

Output:

```
{
  "ShardIterator": "AAAAAAAAAAFEvJjIYI+3jw/4aqqH9FifJ+n48XWTh/
  IFIsbILP6o5eDueD39NXNBfpZ10WL5K6ADXk8w+5H+Qhd9cFA9k268CPXCz/kebq1TGYI7Vy
  +1UkA9BuN3xvATxMBGxRY3zYK05gqgvaIRn9408SqeEqwhigwZxNWxID3Ej7YYYcxQi8Q/fIrCjGAy/
  n2r5Z9G864YpWDFn9upNNQAR/ii0WKs"
}
```

Per ulteriori informazioni, consulta [Developing Consumer Using the Kinesis API Data Streams AWS SDK with the](#) for Java nella Amazon Kinesis Data Streams Developer Guide.

- Per i API dettagli, consulta [GetShardIterator](#) Command Reference.AWS CLI

## increase-stream-retention-period

Il seguente esempio di codice mostra come utilizzare `increase-stream-retention-period`.

## AWS CLI

Per aumentare il periodo di conservazione del flusso di dati

L'increase-stream-retention-period esempio seguente aumenta il periodo di conservazione (il periodo di tempo in cui i record di dati sono accessibili dopo essere stati aggiunti allo stream) del flusso specificato a 168 ore.

```
aws kinesis increase-stream-retention-period \  
  --stream-name samplestream \  
  --retention-period-hours 168
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Modifica del periodo di conservazione dei dati](#) nella Amazon Kinesis Data Streams Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [IncreaseStreamRetentionPeriodReference](#).

## list-shards

Il seguente esempio di codice mostra come utilizzare `list-shards`.

## AWS CLI

Per elencare gli shard in un flusso di dati

L'list-shards esempio seguente elenca tutti gli shard nello stream specificato a partire dallo shard il cui ID segue immediatamente quello specificato di. `exclusive-start-shard-id shardId-000000000000`

```
aws kinesis list-shards \  
  --stream-name samplestream \  
  --exclusive-start-shard-id shardId-000000000000
```

Output:

```
{  
  "Shards": [  
    {  
      "ShardId": "shardId-000000000001",
```

```

    "HashKeyRange": {
      "StartingHashKey": "113427455640312821154458202477256070485",
      "EndingHashKey": "226854911280625642308916404954512140969"
    },
    "SequenceNumberRange": {
      "StartingSequenceNumber":
"49600871682979337187563555549332609155523708941634633746"
    }
  },
  {
    "ShardId": "shardId-000000000002",
    "HashKeyRange": {
      "StartingHashKey": "226854911280625642308916404954512140970",
      "EndingHashKey": "340282366920938463463374607431768211455"
    },
    "SequenceNumberRange": {
      "StartingSequenceNumber":
"49600871683001637932762086172474144873796357303140614178"
    }
  }
]
}

```

Per ulteriori informazioni, consulta [Listing Shards](#) nella Amazon Kinesis Data Streams Developer Guide.

- Per API i dettagli, consulta Command [ListShards](#)Reference AWS CLI .

## list-streams

Il seguente esempio di codice mostra come utilizzare `list-streams`.

### AWS CLI

Per elencare i flussi di dati

L'`list-streams`esempio seguente elenca tutti i flussi di dati attivi nell'account e nella regione correnti.

```
aws kinesis list-streams
```

Output:

```
{
  "StreamNames": [
    "samplestream",
    "samplestream1"
  ]
}
```

Per ulteriori informazioni, consulta [Listing Streams](#) nella Amazon Kinesis Data Streams Developer Guide.

- Per API i dettagli, consulta Command [ListStreams](#)Reference AWS CLI .

## list-tags-for-stream

Il seguente esempio di codice mostra come utilizzare `list-tags-for-stream`.

### AWS CLI

Per elencare i tag per un flusso di dati

L'`list-tags-for-stream`esempio seguente elenca i tag allegati al flusso di dati specificato.

```
aws kinesis list-tags-for-stream \
  --stream-name samplestream
```

Output:

```
{
  "Tags": [
    {
      "Key": "samplekey",
      "Value": "example"
    }
  ],
  "HasMoreTags": false
}
```

Per ulteriori informazioni, consulta [Tagging Your Streams](#) nella Amazon Kinesis Data Streams Developer Guide.

- Per i API dettagli, consulta [ListTagsForStream](#)Command Reference.AWS CLI

## merge-shards

Il seguente esempio di codice mostra come utilizzare `merge-shards`.

### AWS CLI

Per unire frammenti

L'esempio seguente unisce due shard adiacenti con `shardId-000000000000` e `shardId-000000000001` nel flusso IDs di dati specificato e li combina in un unico shard.

```
aws kinesis merge-shards \  
  --stream-name samplestream \  
  --shard-to-merge shardId-000000000000 \  
  --adjacent-shard-to-merge shardId-000000000001
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Merging Two Shards](#) nella Amazon Kinesis Data Streams Developer Guide.

- Per i API dettagli, consulta [MergeShardsCommand Reference](#).AWS CLI

## put-record

Il seguente esempio di codice mostra come utilizzare `put-record`.

### AWS CLI

Per scrivere un record in un flusso di dati

L'esempio seguente scrive un singolo record di dati nel flusso di dati specificato utilizzando la chiave di partizione specificata.

```
aws kinesis put-record \  
  --stream-name samplestream \  
  --data sampledatarecord \  
  --partition-key samplepartitionkey
```

Output:

```
{
```



```

    "ShardId": "shardId-000000000009",
    "SequenceNumber": "49600902273357540915989931256901506243878407835297513618",
    "EncryptionType": "KMS"
  }

```

Per ulteriori informazioni, consulta [Developing Producers Using the Amazon Kinesis API Data Streams AWS SDK with the](#) for Java nella Amazon Kinesis Data Streams Developer Guide.

- Per i API dettagli, consulta [PutRecord](#) Command Reference.AWS CLI

## put-records

Il seguente esempio di codice mostra come utilizzare `put-records`.

### AWS CLI

Per scrivere più record in un flusso di dati

L'`put-records`esempio seguente scrive un record di dati utilizzando la chiave di partizione specificata e un altro record di dati utilizzando una chiave di partizione diversa in una singola chiamata.

```

aws kinesis put-records \
  --stream-name samplestream \
  --
records Data=blob1,PartitionKey=partitionkey1 Data=blob2,PartitionKey=partitionkey2

```

Output:

```

{
  "FailedRecordCount": 0,
  "Records": [
    {
      "SequenceNumber":
"49600883331171471519674795588238531498465399900093808706",
      "ShardId": "shardId-000000000004"
    },
    {
      "SequenceNumber":
"49600902273357540915989931256902715169698037101720764562",
      "ShardId": "shardId-000000000009"
    }
  ]
}

```

```

    }
  ],
  "EncryptionType": "KMS"
}

```

Per ulteriori informazioni, consulta [Developing Producers Using the Amazon Kinesis API Data Streams AWS SDK with the](#) for Java nella Amazon Kinesis Data Streams Developer Guide.

- Per i API dettagli, consulta [PutRecords](#) Command Reference.AWS CLI

## register-stream-consumer

Il seguente esempio di codice mostra come utilizzare `register-stream-consumer`.

### AWS CLI

Per registrare un consumatore di flussi di dati

L'`register-stream-consumer` esempio seguente registra un consumatore chiamato `KinesisConsumerApplication` con il flusso di dati specificato.

```

aws kinesis register-stream-consumer \
  --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \
  --consumer-name KinesisConsumerApplication

```

Output:

```

{
  "Consumer": {
    "ConsumerName": "KinesisConsumerApplication",
    "ConsumerARN": "arn:aws:kinesis:us-west-2:123456789012:stream/samplestream/consumer/KinesisConsumerApplication:1572383852",
    "ConsumerStatus": "CREATING",
    "ConsumerCreationTimestamp": 1572383852.0
  }
}

```

Per ulteriori informazioni, consulta [Developing Consumer with Enhanced Fan-Out Using the Kinesis Data Streams nella Amazon Kinesis Data API Streams Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [RegisterStreamConsumer](#) AWS CLI

## remove-tags-from-stream

Il seguente esempio di codice mostra come utilizzare `remove-tags-from-stream`.

### AWS CLI

Per rimuovere i tag da un flusso di dati

L'esempio seguente rimuove il tag con la chiave specificata dal flusso di dati specificato.

```
aws kinesis remove-tags-from-stream \  
  --stream-name samplestream \  
  --tag-keys samplekey
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Your Streams](#) nella Amazon Kinesis Data Streams Developer Guide.

- Per i API dettagli, consulta [RemoveTagsFromStream](#) Command Reference.AWS CLI

## split-shard

Il seguente esempio di codice mostra come utilizzare `split-shard`.

### AWS CLI

Per dividere i frammenti

L'esempio seguente divide lo shard specificato in due nuovi shard utilizzando una nuova chiave hash iniziale pari a 10.

```
aws kinesis split-shard \  
  --stream-name samplestream \  
  --shard-to-split shardId-000000000000 \  
  --new-starting-hash-key 10
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Splitting a Shard](#) nella Amazon Kinesis Data Streams Developer Guide.

- Per i API dettagli, consulta [SplitShard](#) Command Reference.AWS CLI

## start-stream-encryption

Il seguente esempio di codice mostra come utilizzare `start-stream-encryption`.

### AWS CLI

Per abilitare la crittografia del flusso di dati

L'`start-stream-encryption` esempio seguente abilita la crittografia lato server per il flusso specificato, utilizzando la chiave specificata 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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Protezione dei dati in Amazon Kinesis Data Streams nella Amazon Kinesis](#) Data Streams Developer Guide.

- Per i API dettagli, consulta [StartStreamEncryption](#) Command Reference.AWS CLI

## stop-stream-encryption

Il seguente esempio di codice mostra come utilizzare `stop-stream-encryption`.

### AWS CLI

Per disabilitare la crittografia del flusso di dati

L'`stop-stream-encryption` esempio seguente disattiva la crittografia lato server per il flusso specificato, utilizzando la chiave specificata. 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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Protezione dei dati in Amazon Kinesis Data Streams nella Amazon Kinesis Data Streams Developer Guide](#).

- Per i API dettagli, consulta [StopStreamEncryption](#) Command Reference.AWS CLI

## update-shard-count

Il seguente esempio di codice mostra come utilizzare `update-shard-count`.

### AWS CLI

Per aggiornare il conteggio degli shard in un flusso di dati

L'`update-shard-count`esempio seguente aggiorna il conteggio degli shard del flusso di dati specificato a 6. Questo esempio utilizza un ridimensionamento uniforme, che crea frammenti di dimensioni uguali.

```
aws kinesis update-shard-count \  
  --stream-name samplestream \  
  --scaling-type UNIFORM_SCALING \  
  --target-shard-count 6
```

Output:

```
{  
  "StreamName": "samplestream",  
  "CurrentShardCount": 3,  
  "TargetShardCount": 6  
}
```

Per ulteriori informazioni, consulta [Resharding a Stream nella Amazon Kinesis Data Streams Developer Guide](#).

- Per i API dettagli, consulta [UpdateShardCount](#) Command Reference.AWS CLI

## AWS KMS esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS KMS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **cancel-key-deletion**

Il seguente esempio di codice mostra come utilizzare `cancel-key-deletion`.

AWS CLI

Per annullare l'eliminazione pianificata di una KMS chiave gestita dal cliente

L'`cancel-key-deletion` esempio seguente annulla l'eliminazione pianificata di una KMS chiave gestita dal cliente.

```
aws kms cancel-key-deletion \  
--key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```
{  
  "KeyId": "arn:aws:kms:us-  
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"  
}
```

Quando il `cancel-key-deletion` comando ha esito positivo, l'eliminazione pianificata viene annullata. Tuttavia, lo stato della KMS chiave è `Disabled`, quindi non è possibile utilizzarla nelle operazioni KMS crittografiche. Per ripristinarne la funzionalità, usa il `enable-key` comando.

Per ulteriori informazioni, consulta [Pianificazione e annullamento dell'eliminazione delle chiavi](#) nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere [CancelKeyDeletion](#) in AWS CLI Command Reference.

## connect-custom-key-store

Il seguente esempio di codice mostra come utilizzare `connect-custom-key-store`.

### AWS CLI

Per connettere un archivio di chiavi personalizzato

L'esempio seguente riconnette l'archivio di chiavi personalizzato specificato. È possibile utilizzare un comando come questo per connettere un archivio chiavi personalizzato per la prima volta o per ricollegare un archivio chiavi che era disconnesso.

È possibile utilizzare questo comando per connettere un archivio di HSM chiavi AWS nel cloud o un archivio di chiavi esterno.

```
aws kms connect-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0
```

Questo comando non restituisce alcun output. Per verificare che il comando sia stato efficace, usa il `describe-custom-key-stores` comando.

Per informazioni sulla connessione di un archivio di HSM chiavi AWS Cloud, consulta [Connessione e disconnessione di un archivio di HSM chiavi AWS Cloud](#) nella AWS Key Management Service Developer Guide.

Per informazioni sulla connessione di un archivio di chiavi esterno, consulta [Connessione e disconnessione di un archivio di chiavi esterno](#) nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere [ConnectCustomKeyStore](#) in AWS CLI Command Reference.

## create-alias

Il seguente esempio di codice mostra come utilizzare `create-alias`.

### AWS CLI

Per creare un alias per una chiave KMS

Il `create-alias` comando seguente crea un alias denominato `example-alias` per la KMS chiave identificata dall'ID della chiave. `1234abcd-12ab-34cd-56ef-1234567890ab`

I nomi `alias` devono iniziare con `alias/`. Non utilizzare `alias` che iniziano con `alias/aws`. Questi nomi sono riservati a AWS.

```
aws kms create-alias \  
  --alias-name alias/example-alias \  
  --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non restituisce alcun output. Per vedere il nuovo `alias`, usa il `list-aliases` comando.

Per ulteriori informazioni, vedere [Using alias](#) nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere [CreateAlias](#) in AWS CLI Command Reference.

## create-custom-key-store

Il seguente esempio di codice mostra come utilizzare `create-custom-key-store`.

### AWS CLI

Esempio 1: creare un archivio di HSM chiavi AWS nel cloud

L'esempio seguente crea un archivio di HSM chiavi AWS Cloud supportato da un HSM cluster AWS Cloud utilizzando i parametri richiesti. Puoi anche aggiungere il `custom-key-store-type` parameter with the default value: `AWS_CLOUDHSM`.

Per specificare l'input del file per il `trust-anchor-certificate` comando in AWS CLI, è necessario il `file://` prefisso.

```
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
```

Output:

```
{  
  "CustomKeyId": cks-1234567890abcdef0  
}
```



Per ulteriori informazioni, consulta [Creating an AWS Cloud HSM key store](#) nella AWS Key Management Service Developer Guide.

Esempio 2: creare un archivio di chiavi esterno con connettività endpoint pubblica

L'create-custom-key-storeesempio seguente crea un key store esterno (XKS) con cui comunica AWS KMS tramite Internet.

In questo esempio, XksProxyUriPath utilizza un prefisso opzionale di. example-prefix

NOTE: Se utilizzate la AWS CLI versione 1.0, eseguite il comando seguente prima di specificare un parametro con un HTTPS valore HTTP o, ad esempio, il XksProxyUriEndpoint parametro.

```
aws configure set cli_follow_urlparam false
```

Altrimenti, AWS CLI la versione 1.0 sostituisce il valore del parametro con il contenuto trovato a quell'URLindirizzo.

```
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=DXjSUawneL2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

Output:

```
{  
  "CustomKeyId": cks-2234567890abcdef0  
}
```

Per ulteriori informazioni, vedere [Creazione di un archivio di chiavi esterno](#) nella AWS Key Management Service Developer Guide.

Esempio 3: creare un archivio di chiavi esterno con connettività al servizio VPC endpoint

L'create-custom-key-storeesempio seguente crea un key store esterno (XKS) che utilizza un servizio di VPC endpoint Amazon con AWS KMS cui comunicare.

NOTE: Se utilizzi la AWS CLI versione 1.0, esegui il comando seguente prima di specificare un parametro con un HTTPS valore HTTP o, ad esempio, il XksProxyUriEndpoint parametro.

```
aws configure set cli_follow_urlparam false
```

Altrimenti, AWS CLI la versione 1.0 sostituisce il valore del parametro con il contenuto trovato a quell'URLindirizzo.

```
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=DXjSUawne12fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

Output:

```
{
  "CustomKeyId": cks-3234567890abcdef0
}
```

Per ulteriori informazioni, vedere [Creazione di un archivio di chiavi esterno](#) nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere [CreateCustomKeyStorein](#) AWS CLI Command Reference.

## create-grant

Il seguente esempio di codice mostra come utilizzare `create-grant`.

### AWS CLI

Per creare una sovvenzione

L'`create-grant`esempio seguente crea una concessione che consente all'`exampleUser`utente di utilizzare il `decrypt` comando sulla KMS chiave di `1234abcd-12ab-34cd-56ef-1234567890ab` esempio. Il preside uscente è il `adminRole`

ruolo. La concessione utilizza il vincolo `EncryptionContextSubset` grant per consentire questa autorizzazione solo quando il contesto di crittografia nella `decrypt` richiesta include la `"Department": "IT"` coppia chiave-valore.

```
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
```

Output:

```
{  
  "GrantId": "1a2b3c4d2f5e69f440bae30eaec9570bb1fb7358824f9ddf4a1aa5a0dab1a59b2",  
  "GrantToken": "<grant token here>"  
}
```

Per visualizzare informazioni dettagliate sulla concessione, utilizzare il comando `list-grants`

Per ulteriori informazioni, consulta [Grants AWS KMS nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [CreateGrant](#) in AWS CLI Command Reference.

## create-key

Il seguente esempio di codice mostra come utilizzare `create-key`.

### AWS CLI

Esempio 1: creare una KMS chiave gestita dal cliente in AWS KMS

L'esempio seguente crea una chiave di crittografia KMS simmetrica.

Per creare la KMS chiave di base, una chiave di crittografia simmetrica, non è necessario specificare alcun parametro. I valori predefiniti per tali parametri creano una chiave di crittografia simmetrica.

Poiché questo comando non specifica una politica di chiave, la chiave ottiene la politica di KMS chiave [predefinita per le chiavi create](#) a livello di codice. KMS Per visualizzare la politica chiave,

utilizzare il `get-key-policy` comando. Per modificare la politica chiave, usa il `put-key-policy` comando.

```
aws kms create-key
```

Il `create-key` comando restituisce i metadati della chiave, incluso l'ID della chiave e ARN della nuova KMS chiave. È possibile utilizzare questi valori per identificare la KMS chiave in altre AWS KMS operazioni. L'output non include i tag. Per visualizzare i tag di una KMS chiave, utilizzare il `list-resource-tags` command.

Output:

```
{
  "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"
    ]
  }
}
```

Nota: il `create-key` comando non consente di specificare un alias. Per creare un alias per la nuova KMS chiave, usa il `create-alias` comando.

Per ulteriori informazioni, vedere [Creating keys nella AWS Key Management Service Developer Guide](#).

Esempio 2: creare una RSA KMS chiave asimmetrica per la crittografia e la decrittografia

L'create-keyesempio seguente crea una KMS chiave che contiene una coppia di chiavi asimmetrica per RSA la crittografia e la decrittografia.

```
aws kms create-key \  
  --key-spec RSA_4096 \  
  --key-usage ENCRYPT_DECRYPT
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Asymmetric keys nella AWS KMS Key Management Service Developer Guide](#).AWS

Esempio 3: creare una chiave a curva ellittica asimmetrica per la firma e la verifica KMS

Per creare una chiave asimmetrica che contenga una coppia di KMS chiavi asimmetrica ellittica curva (ECC) per la firma e la verifica. Il --key-usage parametro è obbligatorio anche se SIGN\_VERIFY è l'unico valore valido per le chiavi. ECC KMS

```
aws kms create-key \  
  --key-spec ECC_256 \  
  --key-usage SIGN_VERIFY
```

```
--key-spec ECC_NIST_P521 \  
--key-usage SIGN_VERIFY
```

Output:

```
{  
  "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"  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta la sezione [Chiavi asimmetriche AWS KMS nella AWS Key Management Service Developer Guide](#).

Esempio 4: Per creare una chiave HMAC KMS

L'create-keyesempio seguente crea una chiave a 384 bit HMACKMS. Il GENERATE\_VERIFY\_MAC valore del --key-usage parametro è obbligatorio anche se è l'unico valore valido per HMAC KMS le chiavi.

```
aws kms create-key \  
--key-spec HMAC_384 \  
--key-usage GENERATE_VERIFY_MAC
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [HMACle chiavi AWS KMS nella AWS Key Management Service Developer Guide](#).

Esempio 4: Per creare una chiave primaria KMS multiregionale

L'create-keyesempio seguente crea una chiave di crittografia simmetrica primaria multiregione. Poiché i valori predefiniti per tutti i parametri creano una chiave di crittografia simmetrica, per questa chiave è necessario solo il --multi-region parametro. KMS Nella AWS CLI, per indicare che un parametro booleano è vero, è sufficiente specificare il nome del parametro.

```
aws kms create-key \
  --multi-region
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Asymmetric keys AWS KMS nella AWS Key Management Service Developer Guide](#).

Esempio 5: creare una KMS chiave per il materiale chiave importato

L'create-keyesempio seguente crea una KMS chiave senza materiale chiave. Una volta completata l'operazione, è possibile importare il proprio materiale chiave nella KMS chiave. Per creare questa KMS chiave, impostate il `--origin` parametro su `EXTERNAL`.

```

aws kms create-key \
  --origin EXTERNAL

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Importazione di materiale chiave nelle AWS KMS chiavi nella AWS Key Management Service Developer Guide](#).

Esempio 6: per creare una KMS chiave in un archivio di HSM chiavi AWS Cloud

L'create-keyesempio seguente crea una KMS chiave nell'archivio di HSM chiavi AWS Cloud specificato. L'operazione crea la KMS chiave e i relativi metadati AWS KMS e crea il materiale chiave nel HSM cluster AWS Cloud associato all'archivio di chiavi personalizzato. I parametri --custom-key-store-id e --origin sono obbligatori.

```

aws kms create-key \
  --origin AWS_CLOUDHSM \
  --custom-key-store-id cks-1234567890abcdef0

```

Output:

```

{
  "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",
    "CustomKeyStoreId": "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"
  }
}

```

Per ulteriori informazioni, consulta [AWS Cloud HSM key stores](#) nella Key Management Service Developer Guide.AWS

Esempio 7: creare una KMS chiave in un archivio di chiavi esterno

L'create-keyesempio seguente crea una KMS chiave nell'archivio chiavi esterno specificato. I --xks-key-id parametri --custom-key-store-id--origin, e sono obbligatori in questo comando.

Il --xks-key-id parametro specifica l'ID di una chiave di crittografia simmetrica esistente nel gestore di chiavi esterno. Questa chiave funge da materiale chiave esterno per la KMS chiave. Il valore del --origin parametro deve essere EXTERNAL\_KEY\_STORE .Il custom-key-store-id parametro deve identificare un archivio chiavi esterno collegato al relativo proxy di archiviazione chiavi esterno.

```

aws kms create-key \
  --origin EXTERNAL_KEY_STORE \
  --custom-key-store-id cks-9876543210fedcba9 \
  --xks-key-id bb8562717f809024

```

Output:

```
{
```

```
"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"
  }
}
```

Per ulteriori informazioni, consulta la sezione Archivi di [chiavi esterne nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [CreateKey](#) in AWS CLI Command Reference.

## decrypt

Il seguente esempio di codice mostra come utilizzare `decrypt`.

### AWS CLI

Esempio 1: decrittografare un messaggio crittografato con una KMS chiave simmetrica (Linux e macOS)

Il seguente esempio di `decrypt` comando mostra il metodo consigliato per decrittografare i dati con AWS CLI. Questa versione mostra come decrittografare i dati con una chiave simmetrica.

KMS

Fornisci il testo cifrato in un file. Nel valore del `--ciphertext-blob` parametro, usa il `fileb://` prefisso, che indica loro di leggere i dati CLI da un file binario. Se il file non si trova nella directory corrente, digitate il percorso completo del file. Per ulteriori informazioni sulla lettura AWS CLI dei valori dei parametri da un file, consultate [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) nella AWS Command Line Interface User Guide e [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/) nel AWS Command Line Tool Blog .Specificare la KMS chiave per decrittografare il CipherText. Il `--key-id` parametro non è necessario quando si esegue la decrittografia con una chiave simmetrica. KMS AWS KMS può ottenere l'ID della KMS chiave utilizzata per crittografare i dati dai metadati nel testo cifrato. Ma è sempre consigliabile specificare la KMS chiave che si sta utilizzando. Questa pratica garantisce l'utilizzo della KMS chiave desiderata e impedisce di decifrare inavvertitamente un testo cifrato utilizzando una KMS chiave non attendibile. Richiedete l'output di testo non crittografato come valore di testo. Il `--query` parametro indica loro di ottenere solo il valore del campo dall'output. CLI Plaintext Il `--output` parametro restituisce l'output come `text.base64`: decodifica il testo semplice e lo salva in un file. L'esempio seguente invia (|) il valore del Plaintext parametro all'utilità Base64, che lo decodifica. Quindi, reindirizza (>) l'output decodificato al file. `ExamplePlaintext`

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo account. AWS

```
aws kms decrypt \  
  --ciphertext-blob fileb://ExampleEncryptedFile \  
  --key-id 123abcd-12ab-34cd-56ef-1234567890ab \  
  --output text \  
  --query Plaintext | base64 \  
  --decode > ExamplePlaintextFile
```

Questo comando non produce alcun output. L'output del `decrypt` comando viene decodificato in base64 e salvato in un file.

Per ulteriori informazioni, vedete [Decrypt](#) nel Key Management Service Reference.AWS API

Esempio 2: per decrittografare un messaggio crittografato con una chiave simmetrica KMS (prompt dei comandi di Windows)

L'esempio seguente è lo stesso del precedente, tranne per il fatto che utilizza l'`certutil` utilità per decodificare in Base64 i dati in chiaro. Questa procedura richiede due comandi, come illustrato negli esempi seguenti.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo AWS account.

```
aws kms decrypt ^
  --ciphertext-blob fileb://ExampleEncryptedFile ^
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
  --output text ^
  --query Plaintext > ExamplePlaintextFile.base64
```

Esegui il comando `certutil`.

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Output:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

Per ulteriori informazioni, vedete [Decrypt](#) nel AWS Key Management Service API Reference.

Esempio 3: decrittografare un messaggio crittografato con una chiave asimmetrica KMS (Linux e macOS)

Il seguente esempio di `decrypt` comando mostra come decrittografare i dati crittografati con una chiave asimmetrica. RSA KMS

Quando si utilizza una KMS chiave asimmetrica, è obbligatorio il `encryption-algorithm` parametro che specifica l'algoritmo utilizzato per crittografare il testo in chiaro.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID di chiave valido del tuo account. AWS

```
aws kms decrypt \
  --ciphertext-blob fileb://ExampleEncryptedFile \
  --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
```

```
--encryption-algorithm RSAES_OAEP_SHA_256 \  
--output text \  
--query Plaintext | base64 \  
--decode > ExamplePlaintextFile
```

Questo comando non produce alcun output. L'output del decrypt comando viene decodificato in base64 e salvato in un file.

Per ulteriori informazioni, consulta la sezione [Chiavi asimmetriche nella AWS KMS Key Management Service Developer Guide](#).AWS

- Per API i dettagli, consulta [Decrypt](#) in Command Reference.AWS CLI

## delete-alias

Il seguente esempio di codice mostra come utilizzare. delete-alias

### AWS CLI

Per eliminare un AWS KMS alias

L'delete-alias esempio seguente elimina l'alias. alias/example-alias Il nome dell'alias deve iniziare con alias/.

```
aws kms delete-alias \  
--alias-name alias/example-alias
```

Questo comando non produce alcun output. Per trovare l'alias, usa il comando. list-aliases

Per ulteriori informazioni, vedere [Eliminazione di un alias](#) nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere [DeleteAlias](#) in AWS CLI Command Reference.

## delete-custom-key-store

Il seguente esempio di codice mostra come utilizzare delete-custom-key-store.

### AWS CLI

Per eliminare un archivio di chiavi personalizzato

L'`delete-custom-key-store` seguente elimina l'archivio di chiavi personalizzato specificato.

L'eliminazione di un archivio di HSM chiavi AWS Cloud non ha alcun effetto sul cluster Cloud HSM associato. L'eliminazione di un key store esterno non ha alcun effetto sul proxy del key store esterno associato, sul gestore di chiavi esterno o sulle chiavi esterne.

NOTE: Prima di poter eliminare un archivio chiavi personalizzato, è necessario pianificare l'eliminazione di tutte le KMS chiavi nell'archivio chiavi personalizzato e quindi attendere che tali KMS chiavi vengano eliminate. Quindi, è necessario disconnettere l'archivio chiavi personalizzato. Per informazioni su come trovare le KMS chiavi nel tuo archivio di chiavi personalizzato, consulta [Delete an AWS Cloud HSM key store \(API\)](#) nella AWS Key Management Service Developer Guide.

```
delete-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0
```

Questo comando non restituisce alcun output. Per verificare che l'archivio di chiavi personalizzato venga eliminato, utilizzate il `describe-custom-key-stores` comando.

Per informazioni sull'eliminazione di un archivio di HSM chiavi AWS Cloud, consulta [Eliminazione di un archivio di HSM chiavi AWS Cloud](#) nella AWS Key Management Service Developer Guide.

Per informazioni sull'eliminazione di archivi di chiavi esterni, vedere [Eliminazione di un archivio di chiavi esterne nella AWS Key](#) Management Service Developer Guide.

- Per API i dettagli, vedere [DeleteCustomKeyStore](#) in AWS CLI Command Reference.

## **delete-imported-key-material**

Il seguente esempio di codice mostra come utilizzare `delete-imported-key-material`.

AWS CLI

Per eliminare il materiale chiave importato da una KMS chiave

L'`delete-imported-key-material` seguente elimina il materiale chiave che era stato importato in una KMS chiave.

```
aws kms delete-imported-key-material \  
  --key-id   
  --key-alias   
  --key-usage   
  --key-type   
  --key-length   
  --key-usage   
  --key-type   
  --key-length
```

```
--key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output. Per verificare che il materiale chiave sia stato eliminato, utilizzate il `describe-key` comando per cercare uno stato chiave di `PendingImport` o `PendingDeletion`.

Per ulteriori informazioni, vedete [Eliminazione del materiale chiave importato](https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys-delete-key-material.html) < <https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys-delete-key-material.html> > nella Guida per gli sviluppatori del servizio di gestione delle AWS chiavi.

- Per i API dettagli, vedere in Command Reference. [DeleteImportedKeyMaterial](#) AWS CLI

## derive-shared-secret

Il seguente esempio di codice mostra come utilizzare `derive-shared-secret`.

### AWS CLI

Per ricavare un segreto condiviso

L'`derive-shared-secret` esempio seguente ricava un segreto condiviso utilizzando un algoritmo di accordo chiave.

È necessario utilizzare una curva ellittica asimmetrica NIST consigliata (ECC) o una coppia di KMS key pair SM2 (solo China Regions) con il valore `to call. KeyUsage KEY_AGREEMENT DeriveSharedSecret`

```
aws kms derive-shared-secret \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --key-agreement-algorithm ECDH \
  --public-
key "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvH3Yj0wbkLEpU195Cv1cJVjsVNSjwGq3tCLnzXfhVwv
```

Output:

```
{
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "SharedSecret": "MEYCIQCKZLWyTk5runarx6XiAkU9gv31bwP0/pHa
+DXFehzdDwIhANwpsIV2g/9SPWLLsF6p/hiSskuIXMTRwqrMdVKWMTMHG",
  "KeyAgreementAlgorithm": "ECDH",
```



```
"KeyOrigin": "AWS_KMS"  
}
```

Per ulteriori informazioni, vedere [DeriveSharedSecret](#) nel Key Management Service Reference.AWS API

- Per API i dettagli, vedere [DeriveSharedSecret](#) in AWS CLI Command Reference.

## describe-custom-key-stores

Il seguente esempio di codice mostra come utilizzare `describe-custom-key-stores`.

### AWS CLI

Esempio 1: per ottenere dettagli su un archivio di HSM chiavi AWS Cloud

L'esempio seguente mostra i dettagli sull'archivio di HSM chiavi AWS Cloud specificato. Il comando è lo stesso per tutti i tipi di archivi chiavi personalizzati, ma l'output differisce a seconda del tipo di archivio chiavi e, per un archivio chiavi esterno, della relativa opzione di connettività.

Per impostazione predefinita, questo comando visualizza informazioni su tutti gli archivi di chiavi personalizzati nell'account e nella regione. Per visualizzare informazioni su un particolare archivio di chiavi personalizzato, utilizzare il `custom-key-store-id` parametro `custom-key-store-name` o.

```
aws kms describe-custom-key-stores \  
  --custom-key-store-name ExampleCloudHSMKeyStore
```

L'output di questo comando include dettagli utili sull'archivio di HSM chiavi AWS Cloud, incluso lo stato della connessione (`ConnectionState`). Se lo stato della connessione è `FAILED`, l'output include un `ConnectionErrorCode` campo che descrive il problema.

Output:

```
{  
  "CustomKeyStores": [  
    {  
      "CloudHsmClusterId": "cluster-1a23b4cdefg",  
      "ConnectionState": "CONNECTED",
```

```
        "CreationDate": "2022-04-05T14:04:55-07:00",
        "CustomKeyStoreId": "cks-1234567890abcdef0",
        "CustomKeyStoreName": "ExampleExternalKeyStore",
        "TrustAnchorCertificate": "<certificate appears here>"
    }
]
}
```

Per ulteriori informazioni, consulta [Viewing an AWS Cloud HSM key store](#) nella AWS Key Management Service Developer Guide.

Esempio 2: per ottenere dettagli su un archivio di chiavi esterno con connettività endpoint pubblica

L'output seguente mostra i dettagli sull'archivio di chiavi esterno specificato. Il comando è lo stesso per tutti i tipi di archivi chiavi personalizzati, ma l'output differisce a seconda del tipo di archivio chiavi e, per un archivio chiavi esterno, della relativa opzione di connettività.

Per impostazione predefinita, questo comando visualizza informazioni su tutti gli archivi di chiavi personalizzati nell'account e nella regione. Per visualizzare informazioni su un particolare archivio di chiavi personalizzato, utilizzare il `custom-key-store-id` parametro `custom-key-store-name` o.

```
aws kms describe-custom-key-stores \
  --custom-key-store-id cks-9876543210fedcba9
```

L'output di questo comando include dettagli utili sull'archivio chiavi esterno, incluso lo stato della connessione (`ConnectionState`). Se lo stato della connessione è `FAILED`, l'output include un `ConnectionErrorCode` campo che descrive il problema.

Output:

```
{
  "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXKS",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-02T07:48:55-07:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
```

```
        "AccessKeyId": "ABCDE12345670EXAMPLE",
        "Connectivity": "PUBLIC_ENDPOINT",
        "UriEndpoint": "https://myproxy.xks.example.com",
        "UriPath": "/example-prefix/kms/xks/v1"
    }
}
]
```

Per ulteriori informazioni, vedere [Visualizzazione di un archivio di chiavi esterno](#) nella Guida per gli sviluppatori del servizio di gestione delle AWS chiavi.

Esempio 3: per ottenere dettagli su un archivio di chiavi esterno con connettività al servizio VPC endpoint

L'`describe-custom-key-stores` seguente mostra i dettagli sull'archivio di chiavi esterno specificato. Il comando è lo stesso per tutti i tipi di archivi chiavi personalizzati, ma l'output differisce a seconda del tipo di archivio chiavi e, per un archivio chiavi esterno, della relativa opzione di connettività.

Per impostazione predefinita, questo comando visualizza informazioni su tutti gli archivi di chiavi personalizzati nell'account e nella regione. Per visualizzare informazioni su un particolare archivio di chiavi personalizzato, utilizzare il `custom-key-store-id` parametro `custom-key-store-name` o.

```
aws kms describe-custom-key-stores \
  --custom-key-store-id cks-2234567890abcdef0
```

L'output di questo comando include dettagli utili sull'archivio chiavi esterno, incluso lo stato della connessione (`ConnectionState`). Se lo stato della connessione è `FAILED`, l'output include un `ConnectionErrorCode` campo che descrive il problema.

Output:

```
{
  "CustomKeyStores": [
    {
      "CustomKeyId": "cks-3234567890abcdef0",
      "CustomKeyName": "ExampleVPCExternalKeyStore",
      "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"
    }
  }
]
}

```

Per ulteriori informazioni, vedere [Visualizzazione di un archivio di chiavi esterno](#) nella Guida per gli sviluppatori del servizio di gestione delle AWS chiavi.

- Per API i dettagli, vedere [DescribeCustomKeyStores](#) in AWS CLI Command Reference.

## describe-key

Il seguente esempio di codice mostra come utilizzare `describe-key`.

### AWS CLI

Esempio 1: Per trovare informazioni dettagliate su una KMS chiave

L'`describe-key` esempio seguente ottiene informazioni dettagliate sulla chiave AWS gestita per Amazon S3 nell'account e nella regione di esempio. Puoi utilizzare questo comando per trovare dettagli sulle chiavi gestite e sulle chiavi AWS gestite dai clienti.

Per specificare la KMS chiave, utilizzare il `key-id` parametro. Questo esempio utilizza un valore per il nome di un alias, ma è possibile utilizzare un ID di chiave, una chiaveARN, un nome alias o un alias ARN in questo comando.

```

aws kms describe-key \
  --key-id alias/aws/s3

```

Output:

```
{
```

```

    "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"
      ]
    }
  }
}

```

Per ulteriori informazioni, vedere [Viewing keys nella AWS Key Management Service Developer Guide](#).

Esempio 2: per ottenere dettagli su una chiave RSA asimmetrica KMS

L'`describe-key` seguente ottiene informazioni dettagliate su una RSA KMS chiave asimmetrica utilizzata per la firma e la verifica.

```

aws kms describe-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

```

Output:

```

{
  "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"
    ]
  }
}

```

### Esempio 3: per ottenere dettagli su una chiave di replica multiregionale

L'`describe-key` seguente ottiene i metadati per una chiave di replica multiregionale. Questa chiave multiregionale è una chiave di crittografia simmetrica. L'output di un `describe-key` comando per qualsiasi chiave multiregionale restituisce informazioni sulla chiave primaria e su tutte le relative repliche.

```

aws kms describe-key \
  --key-id arn:aws:kms:ap-northeast-1:111122223333:key/  

mrk-1234abcd12ab34cd56ef1234567890ab

```

Output:

```

{
  "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"
        }
      ]
    }
  }
}

```

Esempio 4: per ottenere dettagli su una chiave HMAC KMS

L'`describe-key` seguente ottiene informazioni dettagliate su una HMAC KMS chiave.

```

aws kms describe-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

```

Output:

```
{
  "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
  }
}
```

- Per API i dettagli, vedere [DescribeKey](#) in AWS CLI Command Reference.

## disable-key-rotation

Il seguente esempio di codice mostra come utilizzare `disable-key-rotation`.

### AWS CLI

Per disattivare la rotazione automatica di una KMS chiave

L'esempio seguente `disable-key-rotation` disabilita la rotazione automatica di una KMS chiave gestita dal cliente. Per riattivare la rotazione automatica, utilizzare il `enable-key-rotation` comando.

```
aws kms disable-key-rotation \
  --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output. Per verificare che la rotazione automatica sia disabilitata per la KMS chiave, usa il `get-key-rotation-status` comando.



Per ulteriori informazioni, consulta [Rotating keys nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [DisableKeyRotation](#) in AWS CLI Command Reference.

## **disable-key**

Il seguente esempio di codice mostra come utilizzare `disable-key`.

### AWS CLI

Per disattivare temporaneamente una KMS chiave

L'esempio seguente utilizza il `disable-key` comando per disabilitare una KMS chiave gestita dal cliente. Per riattivare la KMS chiave, utilizzare il `enable-key` comando.

```
aws kms disable-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Enabling and Disabling Keys nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [DisableKey](#) in AWS CLI Command Reference.

## **disconnect-custom-key-store**

Il seguente esempio di codice mostra come utilizzare `disconnect-custom-key-store`.

### AWS CLI

Per disconnettere un archivio di chiavi personalizzato

L'`disconnect-custom-key-store` esempio seguente disconnette un archivio di chiavi personalizzato dal relativo cluster AWS CloudHSM. È possibile disconnettere un archivio di chiavi per risolvere un problema, aggiornarne le impostazioni o impedire che le KMS chiavi del keystore vengano utilizzate nelle operazioni crittografiche.

Questo comando è lo stesso per tutti gli archivi di chiavi personalizzati, inclusi gli archivi di chiavi AWS Cloud e gli archivi di HSM chiavi esterni.

Prima di eseguire questo comando, sostituisci l'ID store chiavi personalizzate di esempio con uno valido.

```
$ aws kms disconnect-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0
```

Questo comando non produce alcun output. Verifica che il comando sia stato efficace, usa il `describe-custom-key-stores` comando.

Per ulteriori informazioni sulla disconnessione di un archivio di HSM chiavi AWS Cloud, consulta [Connessione e disconnessione di un archivio di HSM chiavi AWS Cloud nella AWS Key Management Service Developer Guide](#).

Per ulteriori informazioni sulla disconnessione di un archivio di chiavi esterno, consulta [Connessione e disconnessione di un archivio di chiavi esterno nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [DisconnectCustomKeyStore](#) in AWS CLI Command Reference.

## **enable-key-rotation**

Il seguente esempio di codice mostra come utilizzare `enable-key-rotation`.

### AWS CLI

Per abilitare la rotazione automatica di una KMS chiave

L'esempio seguente abilita la rotazione automatica di una KMS chiave gestita dal cliente con un periodo di rotazione di 180 giorni. La KMS chiave verrà ruotata di un anno (circa 365 giorni) dalla data di completamento di questo comando e successivamente ogni anno.

Il `--key-id` parametro identifica la chiave. KMS Questo esempio utilizza un ARN valore chiave, ma è possibile utilizzare l'ID della chiave o quello ARN della KMS chiave. Il `--rotation-period-in-days` parametro specifica il numero di giorni tra ogni data di rotazione. Specificate un valore compreso tra 90 e 2560 giorni. Se non viene specificato alcun valore, il valore predefinito è 365 giorni.

```
aws kms enable-key-rotation \  
  --key-id   
  --rotation-period-in-days 180
```

```
--key-id arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \  
--rotation-period-in-days 180
```

Questo comando non produce alcun output. Per verificare che la KMS chiave sia abilitata, usa il `get-key-rotation-status` comando.

Per ulteriori informazioni, consulta [Rotating keys nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [EnableKeyRotation](#) in AWS CLI Command Reference.

## enable-key

Il seguente esempio di codice mostra come utilizzare `enable-key`.

### AWS CLI

Per abilitare una KMS chiave

L'esempio seguente abilita una chiave gestita dal cliente. È possibile utilizzare un comando come questo per abilitare una KMS chiave che è stata temporaneamente disabilitata utilizzando il `disable-key` comando. Puoi anche usarlo per abilitare una KMS chiave che è disabilitata perché era stata pianificata per l'eliminazione e l'eliminazione è stata annullata.

Per specificare la KMS chiave, utilizzare il `key-id` parametro. Questo esempio utilizza un valore ID chiave, ma è possibile utilizzare un ID chiave o un ARN valore chiave in questo comando.

Prima di eseguire questo comando, sostituite l'ID della chiave di esempio con uno valido.

```
aws kms enable-key \  
--key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output. Per verificare che la KMS chiave sia abilitata, utilizzate il `describe-key` comando. Visualizza i valori dei campi `Enabled` e `KeyState` e nell'output di `describe-key`.

Per ulteriori informazioni, vedere [Enabling and Disabling Keys nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [EnableKey](#) in AWS CLI Command Reference.

## encrypt

Il seguente esempio di codice mostra come utilizzare `encrypt`.

### AWS CLI

Esempio 1: cifrare il contenuto di un file su Linux o macOS

Il `encrypt` comando seguente illustra il metodo consigliato per crittografare i dati con AWS CLI

```
aws kms encrypt \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --plaintext fileb://ExamplePlaintextFile \
  --output text \
  --query CiphertextBlob | base64 \
  --decode > ExampleEncryptedFile
```

Il comando esegue diverse operazioni:

Utilizza il `--plaintext` parametro per indicare i dati da crittografare. Il valore del parametro deve essere codificato in base 64. Il valore del `plaintext` parametro deve essere codificato in base 64 oppure è necessario utilizzare il `fileb://` prefisso, che indica loro di leggere i dati binari dal file. Se il AWS CLI file non si trova nella directory corrente, digitate il percorso completo del file. Ad esempio: `fileb:///var/tmp/ExamplePlaintextFile` o `fileb://C:\Temp\ExamplePlaintextFile`. Per ulteriori informazioni sulla lettura AWS CLI dei valori dei [parametri da un file, vedere Loading Parameters from a File](#) nella AWS Command Line Interface User Guide e [Best Practices for Local File --query Parameters](#) nel blog AWS --output Command Line Tool. Utilizza i parametri `and` per controllare l'output del comando. Questi parametri estraggono i dati crittografati, chiamati `ciphertext`, dall'output del comando. Per ulteriori informazioni sul controllo dell'output, vedere [Controllo dell'output Output](#) dei comandi nella Guida per l'utente dell'interfaccia a riga di AWS comando. Utilizza l'`base64` utilità per decodificare l'output estratto in dati binari. Il testo cifrato restituito da un `encrypt` comando riuscito è testo con codifica base64. È necessario decodificare questo testo prima di poterlo utilizzare AWS CLI per decrittografarlo. Salva il testo cifrato binario in un file. La parte finale del comando `()` salva il testo cifrato binario in un file per facilitarne la decrittografia. `> ExampleEncryptedFile` Per un comando di esempio che utilizza il per decrittografare i dati, consultate gli esempi di decrittografia.

### AWS CLI

Esempio 2: utilizzo di per AWS CLI crittografare i dati in Windows

Questo esempio è lo stesso del precedente, tranne per il fatto che utilizza lo `certutil` strumento anziché `base64`. Questa procedura richiede due comandi, come illustrato nell'esempio seguente.

```
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
```

### Esempio 3: crittografia con una chiave asimmetrica KMS

Il `encrypt` comando seguente mostra come crittografare il testo in chiaro con una chiave asimmetrica. KMS Il parametro `--encryption-algorithm` è obbligatorio. Come in tutti `encrypt` CLI i comandi, il `plaintext` parametro deve essere codificato in `base64` oppure è necessario utilizzare il `fileb://` prefisso, che indica loro di leggere i dati binari dal file. AWS CLI

```
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
```

Questo comando non produce alcun output.

- [Per i API dettagli, consulta `Encrypt in Command Reference.AWS CLI`](#)

## generate-data-key-pair-without-plaintext

Il seguente esempio di codice mostra come utilizzare `generate-data-key-pair-without-plaintext`.

### AWS CLI

Per generare una coppia di ECC NIST chiavi dati asimmetriche P384

L'`generate-data-key-pair-without-plaintext` esempio seguente richiede una coppia di chiavi ECC NIST P384 da utilizzare all'esterno di AWS

Il comando restituisce una chiave pubblica in chiaro e una copia della chiave privata crittografata con la chiave specificata. KMS Non restituisce una chiave privata in testo normale. È possibile archiviare in modo sicuro la chiave privata crittografata con i dati crittografati e chiamare AWS KMS per decrittografare la chiave privata quando è necessario utilizzarla.

Per richiedere una coppia di chiavi dati asimmetriche ECC NIST P384, utilizzare `key-pair-spec` il parametro con un valore di `ECC_NIST_P384`

La KMS chiave specificata deve essere una chiave di crittografia simmetrica, ovvero una KMS chiave con un valore di `KMSKeySpec SYMMETRIC_DEFAULT`

NOTE: I valori nell'output di questo esempio vengono troncati per essere visualizzati.

```
aws kms generate-data-key-pair-without-plaintext \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --key-pair-spec ECC_NIST_P384
```

Output:

```
{  
  "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdK12aJTzkK6Fbh0tQkM1QJJH3PdtHvS/y  
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSIb3DQEHbqCCB1...",  
  "PublicKey":  
  "MIIBojANBgkqhkiG9w0BAQEFAAO0CAY8AMIIBigKCAYEA3A3eGMyPrivSn7+Ld1JE1oUoQV5HpEuHAVbd0yND  
+NmYDH/mL10SIEuLrzdZ5hrMH4pk83r401...",  
  "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "KeySpec": "ECC_NIST_P384"  
}
```

I `PublicKey` e `PrivateKeyCiphertextBlob` vengono restituiti in formato con codifica base64.

Per ulteriori informazioni, consulta [Data key pairs](#) nella Key Management Service Developer AWS Guide.

- Per API i dettagli, vedere [GenerateDataKeyPairWithoutPlaintext](#) in AWS CLI Command Reference.

## **generate-data-key-pair**

Il seguente esempio di codice mostra come utilizzare `generate-data-key-pair`.

## AWS CLI

Per generare una coppia di chiavi dati RSA asimmetriche a 2048 bit

L'operazione `generate-data-key-pair` seguente richiede una coppia di chiavi RSA dati asimmetriche a 2048 bit da utilizzare all'esterno di AWS. Il comando restituisce una chiave pubblica in testo semplice e una chiave privata in testo semplice per l'uso e l'eliminazione immediati e una copia della chiave privata crittografata con la chiave specificata. KMS È possibile archiviare in modo sicuro la chiave privata crittografata con i dati crittografati.

Per richiedere una coppia di chiavi RSA dati asimmetriche a 2048 bit, utilizzare `key-pair-spec` il parametro con un valore di `RSA_2048`

La KMS chiave specificata deve essere una chiave di crittografia simmetrica, ovvero una KMS chiave con un valore di `KMSKeySpec SYMMETRIC_DEFAULT`

NOTE: I valori nell'output di questo esempio vengono troncati per essere visualizzati.

```
aws kms generate-data-key-pair \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --key-pair-spec RSA_2048
```

Output:

```
{  
  "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdK12aJTzkK6Fbh0tQkM1QJJH3PdtHvS/y  
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSIB3DQEHBqCCB1...",  
  "PrivateKeyPlaintext": "MIIG/  
QIBADANBgkqhkiG9w0BAQEFAASCBUcwggbjAgEAAoIBgQDcDd4YzI  
+u9Kfv4t2UkTWhShBXkekS4cBVt07I0P42ZgMf+YvU5IgS4ut...",  
  "PublicKey":  
  "MIIB0jANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBigKCAyEA3A3eGMyPrvSn7+Ld1JE1oUoQV5HpEuHAVbd0yND  
+NmYDH/mL10SIEuLrcdZ5hrMH4pk83r40l...",  
  "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "KeySpec": "RSA_2048"  
}
```

I `PublicKey`, `PrivateKeyPlaintext`, e `PrivateKeyCiphertextBlob` vengono restituiti in formato con codifica base64.





- Per API i dettagli, vedere [GenerateDataKeyWithoutPlaintext](#) in AWS CLI Command Reference.

## generate-data-key

Il seguente esempio di codice mostra come utilizzare `generate-data-key`.

### AWS CLI

Esempio 1: generare una chiave dati simmetrica a 256 bit

L'esempio seguente richiede una chiave dati simmetrica a 256 bit da utilizzare all'esterno di AWS. Il comando restituisce una chiave di dati in testo semplice per l'uso e l'eliminazione immediati e una copia di tale chiave dati crittografata con la chiave specificata. KMS Puoi archiviare la chiave di dati crittografata in modo sicuro con i dati crittografati.

Per richiedere una chiave dati a 256 bit, utilizzate il `key-spec` parametro con un valore di `AES_256`. Per richiedere una chiave dati a 128 bit, utilizzate il `key-spec` parametro con un valore di `AES_128`. Per tutte le altre lunghezze delle chiavi dati, utilizzate il `number-of-bytes` parametro.

La chiave KMS specificata deve essere una chiave di crittografia simmetrica, ovvero una chiave KMS con un valore specifico della chiave pari a `_`. `SYMMETRIC DEFAULT`

```
aws kms generate-data-key \  
  --key-id alias/ExampleAlias \  
  --key-spec AES_256
```

Output:

```
{  
  "Plaintext": "VdzKNHGzUAzJeRBVY+uUmofUGGiDzyB3+i9fVkh3piw=",  
  "KeyId": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
  "CiphertextBlob":  
    "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIHvcNAQcGoG8wbQIBADBoBgkqhki+YdhV8MrkBQPeac0ReRVNDt9qlEAt+SHgIRF8P0H+7U="
```

La `Plaintext` (chiave dati in chiaro) e la `CiphertextBlob` (chiave dati crittografata) vengono restituite in formato con codifica base64.

Per ulteriori informazioni, consulta [Data keys < https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#data-keys](https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#data-keys) nella Key Management Service Developer Guide.AWS

Esempio 2: generare una chiave dati simmetrica a 512 bit

L'generate-data-keyesempio seguente richiede una chiave dati simmetrica a 512 bit per la crittografia e la decrittografia. Il comando restituisce una chiave di dati in testo semplice per l'uso e l'eliminazione immediati e una copia di tale chiave dati crittografata con la chiave specificata. KMS Puoi archiviare la chiave di dati crittografata in modo sicuro con i dati crittografati.

Per richiedere una lunghezza di chiave diversa da 128 o 256 bit, utilizzate il parametro. number-of-bytes Per richiedere una chiave dati a 512 bit, l'esempio seguente utilizza il number-of-bytes parametro con un valore di 64 (byte).

La KMS chiave specificata deve essere una chiave di crittografia simmetrica, ovvero una KMS chiave con un valore specifico della KMS chiave pari a `_`. SYMMETRIC DEFAULT

NOTE: I valori nell'output di questo esempio vengono troncati per essere visualizzati.

```
aws kms generate-data-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --number-of-bytes 64
```

Output:

```
{
  "CiphertextBlob": "AQIBAHi6LtupRpdK12aJTzkK6Fbh0tQkM1QJJH3PdtHvS/y+hAEnX/
QQNmMwDfg2korNMEc8AAACaDCCAmQGCSqGSiB3DQEHBqCCA1UwggJRAgEAMIICSgYJKoZ...",
  "Plaintext": "ty8Lr0Bk60F07M2Bwt6qbFdNB
+G00ZLtf5MSEb4a13R2UKWG0p06njAwy2n72VRm2m7z/
Pm9Wpbvttz6a4lSo9hgPvKhZ5y6RTm40ovEXiVfBveyX3DQxDzRSwbKDPk/...",
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Plaintext(chiave dati in chiaro) e CiphertextBlob (chiave dati crittografata) vengono restituite in formato con codifica base64.

Per ulteriori informazioni, consulta [Data keys < https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#data-keys](https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#data-keys) nella Key Management Service Developer Guide.AWS

- Per API i dettagli, vedere [GenerateDataKey](#) in AWS CLI Command Reference.

## generate-random

Il seguente esempio di codice mostra come utilizzare `generate-random`.

### AWS CLI

Esempio 1: generare una stringa di byte casuale a 256 bit (Linux o) macOS

L'esempio seguente genera una stringa di byte casuali a 256 bit (32 byte) con codifica in base64. L'esempio decodifica la stringa di byte e la salva in un file casuale.

Quando si esegue questo comando, è necessario utilizzare il `number-of-bytes` parametro per specificare la lunghezza del valore casuale in byte.

Non si specifica una KMS chiave quando si esegue questo comando. La stringa di byte casuale non è correlata a nessuna KMS chiave.

Per impostazione predefinita, AWS KMS genera il numero casuale. Tuttavia, se si specifica un archivio di chiavi personalizzato < <https://docs.aws.amazon.com/kms/latest/developerguide/custom-key-store-overview.html> >, la stringa di byte casuale viene generata nel HSM cluster AWS Cloud associato all'archivio di chiavi personalizzato.

Questo esempio utilizza i seguenti parametri e valori:

Utilizza il `--number-of-bytes` parametro richiesto con un valore pari 32 a per richiedere una stringa da 32 byte (256 bit). Utilizza il `--output` parametro con valore `text` per indirizzare AWS CLI a restituire l'output come testo, anziché. Utilizza il `--query Plaintext` per estrarre il valore della `Plaintext` proprietà dalla response. IT invia (!) l'output del comando all'base64utilità, che decodifica l'output estratto. Utilizza l'operatore di reindirizzamento (>) per salvare la stringa di byte decodificata in File.it utilizza il `JSON ExampleRandom` operatore di reindirizzamento (>) per salvare il testo cifrato binario in un file.

```
aws kms generate-random \  
  --number-of-bytes 32 \  
  --output text \  
  --query Plaintext | base64 --decode > ExampleRandom
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [GenerateRandom](#) nel AWS Key Management Service Reference API

## Esempio 2: generare un numero casuale a 256 bit (prompt dei comandi di Windows)

L'esempio seguente utilizza il `generate-random` comando per generare una stringa di byte casuali a 256 bit (32 byte) con codifica base64. L'esempio decodifica la stringa di byte e la salva in un file casuale. Questo esempio è lo stesso dell'esempio precedente, tranne per il fatto che utilizza l'utilità di Windows `certutil` per decodificare in base64 la stringa di byte casuale prima di salvarla in un file.

Innanzitutto, genera una stringa di byte casuali codificata in base64 e la salva in un file temporaneo, `ExampleRandom.base64`

```
aws kms generate-random \  
  --number-of-bytes 32 \  
  --output text \  
  --query Plaintext > ExampleRandom.base64
```

Poiché l'output del `generate-random` comando viene salvato in un file, questo esempio non produce alcun output.

Ora utilizzate il `certutil -decode` comando per decodificare la stringa di byte codificata in base64 nel file `ExampleRandom.base64`. Quindi, salva la stringa di byte decodificata nel file `ExampleRandom`

```
certutil -decode ExampleRandom.base64 ExampleRandom
```

Output:

```
Input Length = 18  
Output Length = 12  
CertUtil: -decode command completed successfully.
```

Per ulteriori informazioni, vedere [GenerateRandom](#) nel AWS Key Management Service API Reference.

- Per API i dettagli, vedere [GenerateRandom](#) in AWS CLI Command Reference.

## get-key-policy

Il seguente esempio di codice mostra come utilizzare `get-key-policy`.

## AWS CLI

Per copiare una politica chiave da una KMS chiave a un'altra KMS chiave

L'`get-key-policy` esempio seguente ottiene la politica chiave da una KMS chiave e la salva in un file di testo. Quindi, sostituisce la policy di una KMS chiave diversa utilizzando il file di testo come input della policy.

Poiché il `--policy` parametro di `put-key-policy` richiede una stringa, è necessario utilizzare l'`--output text` opzione per restituire l'output come stringa di JSON testo anziché.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [PutKeyPolicy](#) nel AWS KMS API Reference.

- Per API i dettagli, vedere [GetKeyPolicy](#) in AWS CLI Command Reference.

## get-key-rotation-status

Il seguente esempio di codice mostra come utilizzare `get-key-rotation-status`.

### AWS CLI

Per recuperare lo stato di rotazione di una KMS chiave.

L'`get-key-rotation-status` esempio seguente restituisce informazioni sullo stato di rotazione della KMS chiave specificata, incluso se la rotazione automatica è abilitata, il periodo di rotazione e la successiva data di rotazione pianificata. È possibile utilizzare questo comando sulle KMS chiavi gestite dal cliente e sulle KMS chiavi AWS gestite. Tuttavia, tutte le KMS chiavi AWS gestite vengono ruotate automaticamente ogni anno.

```
aws kms get-key-rotation-status \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```
{  
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",  
  "KeyRotationEnabled": true,  
  "NextRotationDate": "2024-02-14T18:14:33.587000+00:00",  
  "RotationPeriodInDays": 365  
}
```

Per ulteriori informazioni, consulta [Rotating keys nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [GetKeyRotationStatus](#) in AWS CLI Command Reference.

## get-parameters-for-import

Il seguente esempio di codice mostra come utilizzare `get-parameters-for-import`.

AWS CLI

Per ottenere gli elementi necessari per importare il materiale chiave in una KMS chiave

L'`get-parameters-for-import` seguente ottiene la chiave pubblica e il token di importazione necessari per importare il materiale chiave in una KMS chiave. Quando utilizzate il `import-key-material` comando, assicuratevi di utilizzare il token di importazione e il materiale chiave crittografati dalla chiave pubblica restituiti nello stesso `get-parameters-for-import` comando. Inoltre, l'algoritmo di wrapping specificato in questo comando deve essere quello utilizzato per crittografare il materiale della chiave con la chiave pubblica.

Per specificare la KMS chiave, utilizzate il `key-id` parametro. Questo esempio utilizza un ID di chiave, ma è possibile utilizzare un ID di chiave o una chiave ARN in questo comando.

```
aws kms get-parameters-for-import \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --wrapping-algorithm RSAES_OAEP_SHA_256 \  
  --wrapping-key-spec RSA_2048
```

Output:

```
{
  "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
}
```

Per ulteriori informazioni, consulta [Scaricare la chiave pubblica e importare il token nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [GetParametersForImport](#) in AWS CLI Command Reference.

## get-public-key

Il seguente esempio di codice mostra come utilizzare `get-public-key`.

### AWS CLI

Esempio 1: scaricare la chiave pubblica di una chiave asimmetrica KMS

L'`get-public-key` seguente scarica la chiave pubblica di una chiave asimmetrica KMS

Oltre a restituire la chiave pubblica, l'output include le informazioni necessarie per utilizzare la chiave pubblica in modo sicuro all'esterno AWS KMS, tra cui l'utilizzo della chiave e gli algoritmi di crittografia supportati.

```
aws kms get-public-key \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```
{
  "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"  
  ]  
}
```

Per ulteriori informazioni sull'utilizzo delle chiavi asimmetriche in AWS KMS, vedere [Utilizzo KMS delle chiavi simmetriche e asimmetriche](#) nel Key Management Service Reference.AWS API

Esempio 2: convertire una chiave pubblica in DER formato (Linux e macOS)

L'`get-public-key`esempio seguente scarica la chiave pubblica di una chiave asimmetrica e KMS la salva in un file. DER

Quando si utilizza il `get-public-key` comando in AWS CLI, restituisce una chiave pubblica X.509 DER con codifica Base64. In questo esempio viene ottenuto il valore della proprietà come testo. `PublicKey` Base64 decodifica `PublicKey` e lo salva nel file. `public_key.der` Il `output` parametro restituisce l'output come testo, anziché. `JSON` Il `--query` parametro ottiene solo la `PublicKey` proprietà, non le proprietà necessarie per utilizzare la chiave pubblica in modo sicuro all'esterno AWS KMS.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID di chiave valido del tuo AWS account.

```
aws kms get-public-key \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --output text \  
  --query PublicKey | base64 --decode > public_key.der
```

Questo comando non produce alcun output.

Per ulteriori informazioni sull'utilizzo delle chiavi asimmetriche in AWS KMS, vedere [Utilizzo KMS delle chiavi simmetriche e asimmetriche](#) nel Key Management Service Reference.AWS API

- Per i dettagli, vedere in Command Reference. API [GetPublicKey](#)AWS CLI

## **import-key-material**

Il seguente esempio di codice mostra come utilizzare `import-key-material`.

AWS CLI

Per importare materiale chiave in una KMS chiave



L'import-key-material esempio seguente carica il materiale chiave in una KMS chiave creata senza materiale chiave. Lo stato chiave della KMS chiave deve essere PendingImport.

Questo comando utilizza materiale chiave crittografato con la chiave pubblica restituita dal get-parameters-for-import comando. Utilizza anche il token di importazione dello stesso get-parameters-for-import comando.

Il expiration-model parametro indica che il materiale chiave scade automaticamente alla data e all'ora specificate dal valid-to parametro. Quando il materiale chiave scade, AWS KMS elimina il materiale chiave, lo stato della chiave cambia Pending import e la KMS KMS chiave diventa inutilizzabile. Per ripristinare la KMS chiave, è necessario reimportare lo stesso materiale chiave. Per utilizzare un materiale chiave diverso, è necessario creare una nuova KMS chiave.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido o una chiave ARN del tuo AWS account.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni sull'importazione di materiale chiave, vedere [Importazione di materiale chiave](#) nella Key Management Service Developer Guide.AWS

- Per API i dettagli, vedere [ImportKeyMaterial](#) in AWS CLI Command Reference.

## list-aliases

Il seguente esempio di codice mostra come utilizzare list-aliases.

### AWS CLI

Esempio 1: per elencare tutti gli alias in un AWS account e in una regione

L'esempio seguente utilizza il list-aliases comando per elencare tutti gli alias nella regione predefinita dell' AWS account. L'output include alias associati alle chiavi gestite e alle KMS chiavi AWS gestite KMS dal cliente.

```
aws kms list-aliases
```

Output:

```
{
  "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"
    },
    ...
  ]
}
```

Esempio 2: per elencare tutti gli alias per una chiave particolare KMS

L'esempio seguente utilizza il `list-aliases` comando e il relativo `key-id` parametro per elencare tutti gli alias associati a una chiave particolare KMS.

Ogni alias è associato a una sola KMS chiave, ma una KMS chiave può avere più alias. Questo comando è molto utile perché la AWS KMS console elenca solo un alias per ogni chiave. KMS Per trovare tutti gli alias di una KMS chiave, è necessario utilizzare il `list-aliases` comando.

Questo esempio utilizza l'ID della KMS chiave per il `--key-id` parametro, ma in questo comando è possibile utilizzare un ID chiave, una chiaveARN, un nome alias o un aliasARN.

```
aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Working with Aliases](#) nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere [ListAliases](#) in AWS CLI Command Reference.

## list-grants

Il seguente esempio di codice mostra come utilizzare `list-grants`.

### AWS CLI

Per visualizzare le sovvenzioni su una chiave AWS KMS

L'esempio seguente mostra tutte le concessioni sulla KMS chiave AWS gestita specificata per Amazon DynamoDB nel tuo account. Questa concessione consente a DynamoDB di utilizzare KMS la chiave per conto dell'utente per crittografare una tabella DynamoDB prima di scriverla su disco. È possibile utilizzare un comando come questo per visualizzare le concessioni relative alle chiavi gestite e alle KMS chiavi AWS gestite dal cliente nell'account e nella regione KMS. AWS

Questo comando utilizza il `key-id` parametro con un ID chiave per identificare la KMS chiave. È possibile utilizzare un ID o una chiave ARN per identificare la KMS chiave. Per ottenere l'ID della

chiave o la chiave ARN di una KMS chiave AWS gestita, usa il `list-aliases` comando `list-keys` o.

```
aws kms list-grants \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

L'output mostra che la concessione autorizza Amazon DynamoDB a utilizzare KMS la chiave per operazioni crittografiche e consente di visualizzare i dettagli sulla chiave `DescribeKey` () e di ritirare KMS le sovvenzioni (). `RetireGrant` Il `EncryptionContextSubset` vincolo limita queste autorizzazioni alle richieste che includono le coppie di contesti di crittografia specificate. Di conseguenza, le autorizzazioni incluse nella concessione sono valide solo per l'account e la tabella DynamoDB specificati.

```
{
  "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"
    }
  ]
}
```

```
}
```

Per ulteriori informazioni, consulta [Grants AWS KMS nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [ListGrants](#) in AWS CLI Command Reference.

## list-key-policies

Il seguente esempio di codice mostra come utilizzare `list-key-policies`.

### AWS CLI

Per ottenere i nomi delle politiche chiave per una KMS chiave

L'`list-key-policies` esempio seguente ottiene i nomi delle politiche chiave per una chiave gestita dal cliente nell'account e nella regione di esempio. È possibile utilizzare questo comando per trovare i nomi delle politiche chiave per le chiavi AWS gestite e le chiavi gestite dal cliente.

Poiché l'unico nome di policy chiave valido è `default`, questo comando non è utile.

Per specificare la KMS chiave, utilizzare il `key-id` parametro. Questo esempio utilizza un valore ID chiave, ma è possibile utilizzare un ID chiave o una chiave ARN in questo comando.

```
aws kms list-key-policies \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```
{  
  "PolicyNames": [  
    "default"  
  ]  
}
```

Per ulteriori informazioni sulle politiche AWS KMS chiave, vedere [Using Key Policies AWS KMS nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [ListKeyPolicies](#) in AWS CLI Command Reference.

## list-key-rotations

Il seguente esempio di codice mostra come utilizzare `list-key-rotations`.

### AWS CLI

Per recuperare informazioni su tutte le rotazioni dei materiali chiave completate

L'`list-key-rotations` esempio seguente elenca le informazioni su tutte le rotazioni del materiale chiave completate per la chiave specificata. KMS

```
aws kms list-key-rotations \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```
{  
  "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  
}
```

Per ulteriori informazioni, vedete [Rotating keys nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [ListKeyRotations](#) in AWS CLI Command Reference.

## list-keys

Il seguente esempio di codice mostra come utilizzare `list-keys`.

## AWS CLI

Per ottenere le KMS chiavi in un account e in una regione

L'`list-keys` seguente ottiene le KMS chiavi in un account e in una regione. Questo comando restituisce sia le chiavi AWS gestite che le chiavi gestite dal cliente.

```
aws kms list-keys
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Viewing Keys](#) nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere [ListKeys](#) in AWS CLI Command Reference.

## list-resource-tags

Il seguente esempio di codice mostra come utilizzare `list-resource-tags`.

## AWS CLI

Per inserire i tag su una KMS chiave

L'`list-resource-tags` seguente ottiene i tag per una KMS chiave. Per aggiungere o sostituire i tag di risorsa sulle KMS chiavi, utilizzate il `tag-resource` comando. L'output mostra che questa KMS chiave ha due tag di risorsa, ognuno dei quali ha una chiave e un valore.

Per specificare la KMS chiave, utilizzate il `key-id` parametro. Questo esempio utilizza un valore ID chiave, ma è possibile utilizzare un ID chiave o una chiave ARN in questo comando.

```
aws kms list-resource-tags \  
--key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```
{  
  "Tags": [  
    {  
      "TagKey": "Dept",  
      "TagValue": "IT"  
    },  
    {  
      "TagKey": "Purpose",  
      "TagValue": "Test"  
    }  
  ],  
  "Truncated": false  
}
```

Per ulteriori informazioni sull'utilizzo dei tag in AWS KMS, consulta [Tagging keys nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [ListResourceTags](#) in AWS CLI Command Reference.

## **list-retirable-grants**

Il seguente esempio di codice mostra come utilizzare `list-retirable-grants`.

AWS CLI

Per visualizzare le sovvenzioni che un preside può concedere in pensione

L'`list-retirable-grants` seguente mostra tutte le sovvenzioni che l'ExampleAdminutente può ritirare sulle KMS chiavi di un account e di una AWS regione. È



possibile utilizzare un comando come questo per visualizzare le sovvenzioni che qualsiasi titolare del conto può ritirare sulle KMS chiavi dell'account e della AWS regione.

Il valore del `retiring-principal` parametro richiesto deve essere l'Amazon Resource Name (ARN) di un account, utente o ruolo.

Non è possibile specificare un servizio per il valore di `retiring-principal` in questo comando, anche se un servizio può essere il principale ritirante. Per trovare le concessioni in cui un particolare servizio è il principale ritirante, usa il comando. `list-grants`

L'output mostra che `ExampleAdmin` l'utente ha il permesso di ritirare le concessioni su due KMS chiavi diverse nell'account e nella regione. Oltre al titolare uscente, l'account è autorizzato a ritirare qualsiasi sovvenzione contenuta nell'account.

```
aws kms list-retirable-grants \  
  --retiring-principal arn:aws:iam::111122223333:user/ExampleAdmin
```

Output:

```
{  
  "Grants": [  
    {  
      "KeyId": "arn:aws:kms:us-  
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",  
      "GrantId":  
"156b69c63cb154aa21f59929fff19760717be8d9d82b99df53e18b94a15a5e88e",  
      "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":
    "8c94d1f12f5e69f440bae30eac9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
    "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
}

```

Per ulteriori informazioni, consulta [Grants AWS KMS nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [ListRetirableGrants](#) in AWS CLI Command Reference.

## put-key-policy

Il seguente esempio di codice mostra come utilizzare `put-key-policy`.

### AWS CLI

Per modificare la politica chiave di una KMS chiave

L'`put-key-policy` esempio seguente modifica la politica chiave per una chiave gestita dal cliente.

Per iniziare, create una policy chiave e salvatela in un JSON file locale. In questo esempio, il file è `key_policy.json`. È inoltre possibile specificare la politica chiave come valore di stringa del `policy` parametro.

La prima dichiarazione di questa politica chiave autorizza l' AWS account a utilizzare IAM le politiche per controllare l'accesso alla KMS chiave. La seconda istruzione fornisce all'test-userutente il permesso di eseguire i `list-keys` comandi `describe-key` and sulla KMS chiave.

Contenuto di `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" : "*"
    }
  ]
}
```

Per identificare la KMS chiave, in questo esempio viene utilizzato l'ID della chiave, ma è possibile utilizzare anche una chiaveARN. Per specificare la politica della chiave, il comando utilizza il `policy` parametro. Per indicare che la politica è contenuta in un file, utilizza il `file://` prefisso richiesto. Questo prefisso è necessario per identificare i file su tutti i sistemi operativi supportati. Infine, il comando utilizza il `policy-name` parametro con un valore `default`. Se non viene specificato alcun nome di `policy`, il valore predefinito è `default`. L'unico valore valido è `default`.

```
aws kms put-key-policy \  
  --policy-name default \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --policy file://key_policy.json
```

Il comando non produce output. Per verificare che il comando sia stato efficace, utilizzare il `get-key-policy` comando. Il comando di esempio seguente ottiene la politica della chiave per la stessa KMS chiave. Il output parametro con un valore di `text` restituisce un formato di testo facile da leggere.

```
aws kms get-key-policy \  
  --policy-name default \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --output text
```

Output:

```
{  
  "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" : "*"   
    }   
  ]  
}
```

Per ulteriori informazioni, vedere [Modifica di una politica AWS chiave](#) nella Key Management Service Developer Guide.

- Per API i dettagli, vedere [PutKeyPolicy](#) in AWS CLI Command Reference.

## re-encrypt

Il seguente esempio di codice mostra come utilizzare `re-encrypt`.

### AWS CLI

Esempio 1: ricrittografare un messaggio crittografato con una KMS chiave simmetrica diversa (Linux e macOS).

Il seguente esempio di `re-encrypt` comando mostra il metodo consigliato per ricrittografare i dati con AWS CLI

Fornite il testo cifrato in un file. Nel valore del `--ciphertext-blob` parametro, utilizzate il `fileb://` prefisso, che indica loro di leggere i dati CLI da un file binario. Se il file non si trova nella directory corrente, digitate il percorso completo del file. Per ulteriori informazioni sulla lettura AWS CLI dei valori dei parametri da un file, consultate [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) nella AWS Command Line Interface User Guide e [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/) nel AWS Command Line Tool Blog .Specificare la KMS chiave sorgente, che decrittografa il CipherText. Il `--source-key-id` parametro non è richiesto quando si esegue la decrittografia con crittografia simmetrica KMSchiavi. AWS KMSpuò ottenere la KMS chiave utilizzata per crittografare i dati dai metadati nel blob di testo cifrato. Ma è sempre consigliabile specificare la chiave che si sta utilizzando. KMS Questa pratica garantisce l'utilizzo della KMS chiave desiderata e impedisce di decrittografare inavvertitamente un testo cifrato utilizzando una KMS chiave non attendibile. Specificate la chiave di destinazione, che cripta nuovamente i dati. Il parametro KMS è sempre obbligatorio. `--destination-key-id` Questo esempio utilizza una chiaveARN, ma è possibile utilizzare qualsiasi identificatore di chiave valido. Richiedere l'output in chiaro come valore di testo. Il `--query` parametro indica loro di CLI ottenere solo il valore del Plaintext campo dall'output. Il `--output` parametro restituisce l'output come text.BASE64-Decode il testo non crittografato e lo salva in un file. L'esempio seguente trasferisce (|) il valore del parametro all'utilità Base64, che lo decodifica. Plaintext Quindi, reindirizza (>) l'output decodificato al file. ExamplePlaintext

Prima di eseguire questo comando, sostituisci la chiave di esempio IDs con identificatori di chiave validi del tuo account. AWS

```
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
```

Questo comando non produce alcun output. L'output del `re-encrypt` comando viene decodificato in base64 e salvato in un file.

Per ulteriori informazioni, vedere `ReEncrypt` < [https://docs.aws.amazon.com/kms/latest/APIReference/API\\_ReEncrypt.html](https://docs.aws.amazon.com/kms/latest/APIReference/API_ReEncrypt.html) nel Key Management Service Reference.AWS API

Esempio 2: Per crittografare nuovamente un messaggio crittografato con una KMS chiave simmetrica diversa (prompt dei comandi di Windows).

Il seguente esempio di `re-encrypt` comando è lo stesso del precedente, tranne per il fatto che utilizza l'`certutil` utilità per decodificare in Base64 i dati di testo in chiaro. Questa procedura richiede due comandi, come illustrato negli esempi seguenti.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo AWS account.

```
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
```

Quindi usa l'`certutil` utilità

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

Output:

```
Input Length = 18  
Output Length = 12  
CertUtil: -decode command completed successfully.
```

Per ulteriori informazioni, vedere [ReEncrypt](https://docs.aws.amazon.com/kms/latest/APIReference/API_ReEncrypt.html) < [https://docs.aws.amazon.com/kms/latest/APIReference/API\\_ReEncrypt.html](https://docs.aws.amazon.com/kms/latest/APIReference/API_ReEncrypt.html) nel AWS Key Management Service API Reference.

- Per API i dettagli, vedere [ReEncrypt](#) in AWS CLI Command Reference.

## retire-grant

Il seguente esempio di codice mostra come utilizzare `retire-grant`.

### AWS CLI

Ritirare una sovvenzione relativa a una chiave master del cliente

L'`retire-grant` esempio seguente elimina una concessione da una KMS chiave.

Il comando di esempio seguente specifica i parametri `grant-id` e `key-id`. Il valore del `key-id` parametro deve essere la chiave ARN della KMS chiave.

```
aws kms retire-grant \
  --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
  --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output. Per confermare che la concessione è stata ritirata, usa il `list-grants` comando.

Per ulteriori informazioni, consulta [Ritiro e revoca delle sovvenzioni](#) nella AWS Key Management Service Developer Guide.

- Per i API dettagli, vedere [RetireGrant](#) in Command Reference.AWS CLI

## revoke-grant

Il seguente esempio di codice mostra come utilizzare `revoke-grant`.

### AWS CLI

Per revocare una concessione su una chiave master del cliente

L'`revoke-grant` esempio seguente elimina una concessione da una chiave. KMS Il comando di esempio seguente specifica i parametri `grant-id` e `key-id`. Il valore del `key-id` parametro può essere l'ID o la chiave ARN della KMS chiave.

```
aws kms revoke-grant \  
  --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output. Per confermare che la concessione è stata revocata, usa il `list-grants` comando.

Per ulteriori informazioni, consulta [Ritiro e revoca delle sovvenzioni](#) nella AWS Key Management Service Developer Guide.

- Per i API dettagli, vedere [RevokeGrant](#) in Command Reference.AWS CLI

## rotate-key-on-demand

Il seguente esempio di codice mostra come utilizzare `rotate-key-on-demand`.

### AWS CLI

Per eseguire la rotazione su richiesta di una chiave KMS

L'esempio seguente avvia immediatamente la rotazione del materiale chiave per la chiave specificata KMS.

```
aws kms rotate-key-on-demand \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Output:

```
{  
  "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"  
}
```

Per ulteriori informazioni, vedere [Come eseguire la rotazione delle chiavi su richiesta nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [RotateKeyOnDemand](#) in AWS CLI Command Reference.

## schedule-key-deletion

Il seguente esempio di codice mostra come utilizzare `schedule-key-deletion`.



## AWS CLI

Per pianificare l'eliminazione di una KMS chiave gestita dal cliente.

L'esempio seguente pianifica l'eliminazione della KMS chiave gestita dal cliente specificata in 15 giorni.

Il `--key-id` parametro identifica la KMS chiave. Questo esempio utilizza un ARN valore chiave, ma è possibile utilizzare l'ID della chiave o la ARN KMS chiave. Il `--pending-window-in-days` parametro specifica la durata del periodo di attesa di 7-30 giorni. Per impostazione predefinita, il periodo di attesa è di 30 giorni. Questo esempio specifica il valore 15, che indica di AWS eliminare definitivamente la KMS chiave 15 giorni dopo il completamento del comando.

```
aws kms schedule-key-deletion \
  --key-id arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \
  --pending-window-in-days 15
```

La risposta include la chiaveARN, lo stato della chiave, il periodo di attesa (`PendingWindowInDays`) e la data di eliminazione in formato Unix. Per visualizzare la data di cancellazione nell'ora locale, usa la AWS KMS console. KMSle chiavi nello stato della `PendingDeletion` chiave non possono essere utilizzate nelle operazioni crittografiche.

```
{
  "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
}
```

Per ulteriori informazioni, vedere [Eliminazione delle chiavi nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [ScheduleKeyDeletion](#) in AWS CLI Command Reference.

## sign

Il seguente esempio di codice mostra come utilizzare `sign`.

## AWS CLI

### Esempio 1: generare una firma digitale per un messaggio

L'esempio seguente genera una firma crittografica per un breve messaggio. L'output del comando include un `Signature` campo codificato in base 64 che è possibile verificare utilizzando il comando `verify`.

È necessario specificare un messaggio da firmare e un algoritmo di firma supportato dalla chiave KMS asimmetrica. Per ottenere gli algoritmi di firma per la tua KMS chiave, usa il comando `describe-key`.

Nella AWS CLI versione 2.0, il valore del `message` parametro deve essere codificato in Base64. In alternativa, è possibile salvare il messaggio in un file e utilizzare il `fileb://` prefisso, che indica loro di leggere i dati binari dal AWS CLI file.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo AWS account. L'ID della chiave deve rappresentare una KMS chiave asimmetrica con un utilizzo chiave di `_SIGN_VERIFY`.

```
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
```

### Output:

```
{
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Signature": "ABCDEFhpyVYyTxbafE74ccSvEJLJr3zuoV1Hfymz4qv+
fxmxNLA7SE1SiF8lHw80fKZZ3bJ...",
  "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

Per ulteriori informazioni sull'utilizzo delle KMS chiavi asimmetriche in AWS KMS, consulta la sezione Chiavi asimmetriche nella [Key Management Service Developer Guide](#). AWS KMS AWS

### Esempio 2: per salvare una firma digitale in un file (Linux e) macOS

L'esempio seguente genera una firma crittografica per un breve messaggio memorizzato in un file locale. Il comando ottiene anche la `Signature` proprietà dalla risposta, Base64-la decodifica e la salva nel file. `ExampleSignature` È possibile utilizzare il file della firma in un `verify` comando che verifica la firma.

Il `sign` comando richiede un messaggio con codifica Base64 e un algoritmo di firma supportato dalla chiave asimmetrica. KMS Per ottenere gli algoritmi di firma supportati dalla chiave, usa il comando. `KMS describe-key`

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo AWS account. L'ID della chiave deve rappresentare una KMS chiave asimmetrica con un utilizzo chiave di `_`. `SIGN VERIFY`

```
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
```

Questo comando non produce alcun output. Questo esempio estrae la `Signature` proprietà dell'output e la salva in un file.

Per ulteriori informazioni sull'utilizzo delle chiavi asimmetriche in AWS KMS, vedete [Asymmetric KMS keys AWS KMS nella](#) Key Management Service Developer Guide.AWS

- [Per i API dettagli, consulta Sign in Command Reference.AWS CLI](#)

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere un tag a una KMS chiave

L'esempio seguente aggiunge `"Purpose": "Test"` e `"Dept": "IT"` contrassegna una KMS chiave gestita dal cliente. È possibile utilizzare tag come questi per etichettare KMS le chiavi e creare categorie di KMS chiavi per le autorizzazioni e il controllo.

Per specificare la KMS chiave, utilizzate il `key-id` parametro. Questo esempio utilizza un valore ID chiave, ma è possibile utilizzare un ID chiave o una chiave ARN in questo comando.

```
aws kms tag-resource \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --tags TagKey='Purpose',TagValue='Test' TagKey='Dept',TagValue='IT'
```

Questo comando non produce alcun output. Per visualizzare i tag su un AWS KMS KMS tasto, usa il `list-resource-tags` comando.

Per ulteriori informazioni sull'utilizzo dei tag in AWS KMS, consulta [Tagging keys nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per eliminare un tag da una KMS chiave

L'`untag-resource` esempio seguente elimina il tag con la "Purpose" chiave da una KMS chiave gestita dal cliente.

Per specificare la KMS chiave, utilizzate il `key-id` parametro. Questo esempio utilizza un valore ID chiave, ma è possibile utilizzare un ID chiave o una chiave ARN in questo comando. Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo AWS account.

```
aws kms untag-resource \  
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \  
  --tag-key 'Purpose'
```

Questo comando non produce alcun output. Per visualizzare i tag su un AWS KMS KMS tasto, usa il `list-resource-tags` comando.

Per ulteriori informazioni sull'utilizzo dei tag in AWS KMS, consulta [Tagging keys nella AWS Key Management Service Developer Guide](#).

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-alias

Il seguente esempio di codice mostra come utilizzare `update-alias`.

### AWS CLI

Per associare un alias a una chiave diversa KMS

L'`update-alias` esempio seguente associa l'alias a una chiave `alias/test-key` diversa. KMS

Il `--alias-name` parametro specifica l'alias. Il valore del nome alias deve iniziare con `alias/`. Il `--target-key-id` parametro specifica la KMS chiave da associare all'alias. Non è necessario specificare la KMS chiave corrente per l'alias.

```
aws kms update-alias \  
  --alias-name alias/test-key \  
  --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

Questo comando non produce alcun output. Per trovare l'alias, usa il `list-aliases` comando.

Per ulteriori informazioni, vedere [Aggiornamento degli alias](#) nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere [UpdateAlias](#) in AWS CLI Command Reference.

## update-custom-key-store

Il seguente esempio di codice mostra come utilizzare `update-custom-key-store`.

### AWS CLI

Esempio 1: modificare il nome descrittivo di un archivio di chiavi personalizzato

L'`update-custom-key-store` esempio seguente modifica il nome dell'archivio chiavi personalizzato. Questo esempio funziona per un HSM key store AWS Cloud o un key store esterno.

Usa il `custom-key-store-id` per identificare l'archivio delle chiavi. Utilizzate il `new-custom-key-store-name` parametro per specificare il nuovo nome descrittivo.

Per aggiornare il nome descrittivo di un archivio HSM chiavi AWS Cloud, devi prima disconnetterlo, ad esempio utilizzando il `disconnect-custom-key-store` comando. È possibile aggiornare il nome descrittivo di un archivio di chiavi esterno mentre è connesso o disconnesso. Per trovare lo stato di connessione del tuo archivio chiavi personalizzato, usa il `describe-custom-key-store` comando.

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --new-custom-key-store-name ExampleKeyStore
```

Questo comando non restituisce alcun dato. Per verificare che il comando abbia funzionato, usa un `describe-custom-key-stores` comando.

Per ulteriori informazioni sull'aggiornamento di un archivio di HSM chiavi AWS Cloud, consulta [Modifica delle impostazioni di AWS Cloud HSM key store](#) nella AWS Key Management Service Developer Guide.

Per ulteriori informazioni sull'aggiornamento di un archivio di chiavi esterno, consulta [Modifica delle proprietà dell'archivio di chiavi esterne](#) nella AWS Key Management Service Developer Guide.

Esempio 2: per modificare la password `kmsuser` di un archivio di chiavi AWS Cloud HSM

L'`update-custom-key-store` esempio seguente aggiorna il valore della `kmsuser` password con la password corrente per il HSM cluster `kmsuser` in the Cloud associato all'archivio di chiavi specificato. Questo comando non modifica la `kmsuser` password del cluster. Indica solo AWS KMS la password corrente. Se KMS non dispone della `kmsuser` password corrente, non può connettersi al AWS Cloud HSM Key Store.

NOTE: Prima di aggiornare un archivio di HSM chiavi AWS Cloud, è necessario disconnetterlo. Utilizza il comando `disconnect-custom-key-store`. Al termine del comando, puoi ricollegare l'archivio di chiavi AWS CloudHSM. Utilizza il comando `connect-custom-key-store`.

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --key-store-password ExamplePassword
```

Questo comando non restituisce alcun output. Per verificare che la modifica sia stata effettiva, usa un `describe-custom-key-stores` comando.

Per ulteriori informazioni sull'aggiornamento di un archivio di HSM chiavi AWS Cloud, consulta [Modifica delle impostazioni di AWS Cloud HSM key store](#) nella AWS Key Management Service Developer Guide.

Esempio 3: per modificare il HSM cluster AWS Cloud di un archivio di HSM chiavi AWS Cloud

L'esempio seguente modifica il HSM cluster AWS Cloud associato a un archivio di HSM chiavi AWS Cloud in un cluster correlato, ad esempio un backup diverso dello stesso cluster.

NOTE: Prima di aggiornare un archivio di HSM chiavi AWS Cloud, è necessario disconnetterlo. Utilizza il comando `disconnect-custom-key-store`. Al termine del comando, puoi ricollegare l'archivio di chiavi AWS Cloud HSM. Utilizza il comando `connect-custom-key-store`.

```
aws kms update-custom-key-store \  
  --custom-key-store-id cks-1234567890abcdef0 \  
  --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

Questo comando non restituisce alcun output. Per verificare che la modifica sia stata effettiva, usa un `describe-custom-key-stores` comando.

Per ulteriori informazioni sull'aggiornamento di un archivio di HSM chiavi AWS Cloud, consulta [Modifica delle impostazioni di AWS Cloud HSM key store](#) nella AWS Key Management Service Developer Guide.

Esempio 4: per modificare le credenziali di autenticazione proxy di un key store esterno

L'esempio seguente aggiorna le credenziali di autenticazione proxy per l'archivio di chiavi esterno. È necessario specificare `raw-secret-access-key` sia il `cheaccess-key-id`, anche se si sta modificando solo uno dei valori. È possibile utilizzare questa funzionalità per correggere una credenziale non valida o per modificare la credenziale quando il proxy dell'archivio chiavi esterno la ruota.

Stabilisci la credenziale di autenticazione proxy per il tuo archivio di chiavi AWS KMS esterno. Quindi usa questo comando per fornire la credenziale a. AWS KMS AWS KMSutilizza questa credenziale per firmare le sue richieste al proxy esterno dell'archivio chiavi.

È possibile aggiornare le credenziali di autenticazione del proxy mentre l'archivio chiavi esterno è connesso o disconnesso. Per trovare lo stato di connessione del tuo archivio chiavi personalizzato, usa il `describe-custom-key-store` comando.

```
aws kms update-custom-key-store \  
  --raw-secret-access-key raw-secret-access-key \  
  --cheaccess-key-id cheaccess-key-id \  
  --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

```
--custom-key-store-id cks-1234567890abcdef0 \  
--xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,  
RawSecretAccessKey=DXjSUawneL2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo=""
```

Questo comando non restituisce alcun output. Per verificare che la modifica sia stata effettiva, utilizzate un `describe-custom-key-stores` comando.

Per ulteriori informazioni sull'aggiornamento di un archivio di chiavi esterno, vedere [Modifica delle proprietà dell'archivio chiavi esterno](#) nella AWS Key Management Service Developer Guide.

Esempio 5: per modificare la connettività proxy di un archivio di chiavi esterno

L'esempio seguente modifica l'opzione di connettività proxy dell'archivio di chiavi esterne dalla connettività degli endpoint pubblici alla connettività del servizio VPC endpoint. Oltre a modificare il `xks-proxy-connectivity` valore, è necessario modificare il `xks-proxy-uri-endpoint` valore in modo che rifletta il DNS nome privato associato al servizio VPC endpoint. È inoltre necessario aggiungere un `xks-proxy-vpc-endpoint-service-name` valore.

NOTE: Prima di aggiornare la connettività proxy di un archivio esterno, è necessario disconnetterlo. Utilizza il comando `disconnect-custom-key-store`. Al termine del comando, è possibile ricollegare l'archivio di chiavi esterno utilizzando il comando `connect-custom-key-store`

```
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"
```

Questo comando non restituisce alcun output. Per verificare che la modifica sia stata effettiva, utilizzate un `describe-custom-key-stores` comando.

Per ulteriori informazioni sull'aggiornamento di un archivio di chiavi esterno, vedere [Modifica delle proprietà dell'archivio chiavi esterno](#) nella AWS Key Management Service Developer Guide.

- Per API i dettagli, vedere [UpdateCustomKeyStore](#) in AWS CLI Command Reference.

## update-key-description

Il seguente esempio di codice mostra come utilizzare `update-key-description`.



## AWS CLI

Esempio 1: aggiungere o modificare una descrizione a una KMS chiave gestita dal cliente

L'`update-key-description` esempio seguente aggiunge una descrizione a una KMS chiave gestita dal cliente. È possibile utilizzare lo stesso comando per modificare una descrizione esistente.

Il `--key-id` parametro identifica la KMS chiave nel comando. Questo esempio utilizza un ARN valore chiave, ma è possibile utilizzare l'ID della chiave o la KMS chiave ARN. Il `--description` parametro specifica la nuova descrizione. Il valore di questo parametro sostituisce l'eventuale descrizione corrente della KMS chiave.

```
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"
```

Questo comando non produce alcun output. Per visualizzare la descrizione di una KMS chiave, utilizzare il `describe-key` comando.

Per ulteriori informazioni, vedere [UpdateKeyDescription](#) nel AWS Key Management Service API Reference.

Esempio 2: Per eliminare la descrizione di una KMS chiave gestita dal cliente

L'`update-key-description` esempio seguente elimina la descrizione di una KMS chiave gestita dal cliente.

Il `--key-id` parametro identifica la KMS chiave nel comando. Questo esempio utilizza un valore ID chiave, ma è possibile utilizzare l'ID della chiave o la KMS chiave. Il `--description` parametro con un valore ARN di stringa vuoto («») elimina la descrizione esistente.

```
aws kms update-key-description \  
  --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \  
  --description ''
```

Questo comando non produce alcun output. Per visualizzare la descrizione di una KMS chiave, utilizzate il comando `describe-key`.

Per ulteriori informazioni, vedere [UpdateKeyDescription](#) nel AWS Key Management Service Reference. API

- Per API i dettagli, vedere [UpdateKeyDescription](#) in AWS CLI Command Reference.

## verify

Il seguente esempio di codice mostra come utilizzare `verify`.

### AWS CLI

Per verificare una firma digitale

L'esempio seguente verifica una firma crittografica per un breve messaggio con codifica Base64. L'ID della chiave, il messaggio, il tipo di messaggio e l'algoritmo di firma devono essere gli stessi utilizzati per firmare il messaggio. La firma specificata non può essere codificata in base 64. Per informazioni sulla decodifica della firma restituita dal comando, consultate gli esempi di `sign` comandi. `sign`

L'output del comando include un `SignatureValid` campo booleano che indica che la firma è stata verificata. Se la convalida della firma fallisce, anche il `verify` comando fallisce.

Prima di eseguire questo comando, sostituisci l'ID della chiave di esempio con un ID chiave valido del tuo AWS account.

```
aws kms verify \
  --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
  --message fileb://EncodedMessage \
  --message-type RAW \
  --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \
  --signature fileb://ExampleSignature
```

Output:

```
{
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "SignatureValid": true,
  "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

Per ulteriori informazioni sull'utilizzo delle chiavi asimmetriche in AWS KMS, vedere [Using asymmetric KMS keys nella Key Management Service Developer Guide](#).AWS

- [Per i API dettagli, consulta Verify in Command Reference.AWS CLI](#)

## Esempi di Lake Formation utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Lake Formation.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **add-lf-tags-to-resource**

Il seguente esempio di codice mostra come utilizzare `add-lf-tags-to-resource`.

AWS CLI

Per allegare uno o più tag LF a una risorsa esistente

L'`add-lf-tags-to-resource`esempio seguente allega un determinato LF-tag alla risorsa della tabella.

```
aws lakeformation add-lf-tags-to-resource \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
```

```
"CatalogId": "123456789111",
"Resource": {
  "Table": {
    "CatalogId": "123456789111",
    "DatabaseName": "tpc",
    "Name": "dl_tpc_promotion"
  }
},
"LFTags": [{
  "CatalogId": "123456789111",
  "TagKey": "usergroup",
  "TagValues": [
    "analyst"
  ]
}]
}
```

Output:

```
{
  "Failures": []
}
```

Per ulteriori informazioni, consulta [Assegnazione di tag LF alle risorse del Data Catalog](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, vedere [AddLfTagsToResource](#) in AWS CLI Command Reference.

## batch-grant-permissions

Il seguente esempio di codice mostra come utilizzare `batch-grant-permissions`.

AWS CLI

Per concedere in blocco le autorizzazioni sulle risorse ai responsabili

L'esempio seguente concede in blocco l'accesso su risorse specifiche ai principali.

```
aws lakeformation batch-grant-permissions \
  --cli-input-json file://input.json
```

## Contenuto di 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": []
}
]
```

Output:

```
{
  "Failures": []
}
```

Per ulteriori informazioni, consulta [Concessione e revoca delle autorizzazioni sulle risorse del Data Catalog](#) nella AWS Lake Formation Developer Guide.

- Per i API dettagli, consulta [BatchGrantPermissions](#) Command Reference.AWS CLI

## batch-revoke-permissions

Il seguente esempio di codice mostra come utilizzare `batch-revoke-permissions`.

### AWS CLI

Per revocare in blocco le autorizzazioni sulle risorse ai responsabili

L'esempio seguente revoca in blocco l'accesso a risorse specifiche ai principali.

```
aws lakeformation batch-revoke-permissions \
  --cli-input-json file://input.json
```

Contenuto di `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"
      ]
    }
  ]
}

```

Output:

```

{
  "Failures": []
}

```

Per ulteriori informazioni, consulta [Concessione e revoca delle autorizzazioni sulle risorse del Data Catalog](#) nella AWS Lake Formation Developer Guide.

- Per i API dettagli, consulta [BatchRevokePermissions](#) Command Reference.AWS CLI

## cancel-transaction

Il seguente esempio di codice mostra come utilizzare `cancel-transaction`.



## AWS CLI

Per annullare una transazione

L'`cancel-transaction`esempio seguente annulla la transazione.

```
aws lakeformation cancel-transaction \  
  --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Leggere e scrivere nel data lake all'interno delle transazioni nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [CancelTransaction AWS CLI Command Reference](#).

## `commit-transaction`

Il seguente esempio di codice mostra come utilizzare `commit-transaction`.

## AWS CLI

Per effettuare una transazione

L'`commit-transaction`esempio seguente esegue il commit della transazione.

```
aws lakeformation commit-transaction \  
  --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

Output:

```
{  
  "TransactionStatus": "committed"  
}
```

Per ulteriori informazioni, consulta [Leggere e scrivere nel data lake all'interno delle transazioni nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [CommitTransaction AWS CLI Command Reference](#).

## create-data-cells-filter

Il seguente esempio di codice mostra come utilizzare `create-data-cells-filter`.

### AWS CLI

Esempio 1: per creare un filtro per celle di dati

L'`create-data-cells-filter` seguente crea un filtro di celle di dati per consentire l'accesso a determinate colonne in base alla condizione della riga.

```
aws lakeformation create-data-cells-filter \  
  --cli-input-json file://input.json
```

Contenuto di `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"  
  }  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Filtraggio dei dati e sicurezza a livello di cella in Lake Formation nella Lake Formation Developer AWS Guide](#).

Esempio 2: per creare un filtro a colonna

L'`create-data-cells-filter` seguente crea un filtro dati per consentire l'accesso a determinate colonne.

```
aws lakeformation create-data-cells-filter \  
  --cli-input-json file://input.json
```

Contenuto di `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"
  }
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Filtraggio dei dati e sicurezza a livello di cella in Lake Formation nella Lake Formation Developer AWS Guide](#).

Esempio 3: per creare un filtro dati con colonne di esclusione

L'`create-data-cells-filter` esempio seguente crea un filtro dati per consentire l'accesso a tutte le colonne tranne quelle menzionate.

```
aws lakeformation create-data-cells-filter \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "TableData": {
    "ColumnWildcard": {
      "ExcludedColumnNames": ["p_channel_details", "p_start_date_sk"]
    },
    "DatabaseName": "tpc",
    "Name": "developer_promotion_excludecolumn",
    "RowFilter": {
      "AllRowsWildcard": {}
    },
    "TableCatalogId": "123456789111",
    "TableName": "dl_tpc_promotion"
  }
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Filtraggio dei dati e sicurezza a livello di cella in Lake Formation nella Lake Formation Developer AWS Guide](#).

- Per API i dettagli, consulta Command [CreateDataCellsFilter](#)Reference AWS CLI .

## **create-lf-tag**

Il seguente esempio di codice mostra come utilizzare `create-lf-tag`.

### AWS CLI

Per creare un tag LF

L'`create-lf-tag`esempio seguente crea un tag LF con il nome e i valori specificati.

```
aws lakeformation create-lf-tag \  
  --catalog-id '123456789111' \  
  --tag-key 'usergroup' \  
  --tag-values ['developer','analyst','campaign']
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing LF-Tags for metadata access control](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consultate Command [CreateLfTag](#)Reference AWS CLI .

## **delete-data-cells-filter**

Il seguente esempio di codice mostra come utilizzare `delete-data-cells-filter`.

### AWS CLI

Per eliminare il filtro delle celle di dati

L'`delete-data-cells-filter`esempio seguente elimina un determinato filtro di celle di dati.

```
aws lakeformation delete-data-cells-filter \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "TableCatalogId": "123456789111",
  "DatabaseName": "tpc",
  "TableName": "dl_tpc_promotion",
  "Name": "developer_promotion"
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Filtraggio dei dati e sicurezza a livello di cella in Lake Formation nella Lake Formation Developer AWS Guide](#).

- Per API i dettagli, consulta Command [DeleteDataCellsFilter](#) Reference AWS CLI .

## delete-lf-tag

Il seguente esempio di codice mostra come utilizzare `delete-lf-tag`.

AWS CLI

Per eliminare la definizione del tag LF

L'`delete-lf-tag` esempio seguente elimina la definizione del tag LF.

```
aws lakeformation delete-lf-tag \
  --catalog-id '123456789111' \
  --tag-key 'usergroup'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing LF-Tags for metadata access control](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consultate Command [DeleteLfTag](#) Reference AWS CLI .

## delete-objects-on-cancel

Il seguente esempio di codice mostra come utilizzare `delete-objects-on-cancel`.

AWS CLI

Per eliminare l'oggetto quando la transazione viene annullata

L'example seguente elimina l'oggetto s3 elencato quando la transazione viene annullata.

```
aws lakeformation delete-objects-on-cancel \  
  --cli-input-json file://input.json
```

Contenuto di `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"  
  }]  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Leggere e scrivere nel data lake all'interno delle transazioni nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [DeleteObjectsOnCancel AWS CLI Command Reference](#).

## deregister-resource

Il seguente esempio di codice mostra come utilizzare `deregister-resource`.

### AWS CLI

Per annullare la registrazione di Data Lake Storage

L'example seguente annulla la registrazione della risorsa come gestita da Lake Formation.

```
aws lakeformation deregister-resource \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123"
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere una posizione Amazon S3 al data lake nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [DeregisterResource AWS CLI Command Reference](#).

## **describe-transaction**

Il seguente esempio di codice mostra come utilizzare `describe-transaction`.

AWS CLI

Per recuperare i dettagli di una transazione

L'`describe-transaction` seguente restituisce i dettagli di una singola transazione.

```
aws lakeformation describe-transaction \
  --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

Output:

```
{
  "TransactionDescription": {
    "TransactionId": "12345972ca8347b89825e33c5774aec4",
    "TransactionStatus": "committed",
    "TransactionStartTime": "2022-08-10T14:29:04.046000+00:00",
    "TransactionEndTime": "2022-08-10T14:29:09.681000+00:00"
  }
}
```

Per ulteriori informazioni, consulta [Leggere e scrivere nel data lake all'interno delle transazioni nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [DescribeTransaction AWS CLI Command Reference](#).

## extend-transaction

Il seguente esempio di codice mostra come utilizzare `extend-transaction`.

### AWS CLI

Per estendere una transazione

L'`extend-transaction` esempio seguente estende la transazione.

```
aws lakeformation extend-transaction \  
  --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Leggere e scrivere nel data lake all'interno delle transazioni nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [ExtendTransaction AWS CLI Command Reference](#).

## get-data-lake-settings

Il seguente esempio di codice mostra come utilizzare `get-data-lake-settings`.

### AWS CLI

Per recuperare le impostazioni AWS del data lake gestite da Lake Formation

L'`get-data-lake-settings` esempio seguente recupera l'elenco degli amministratori del data lake e altre impostazioni del data lake.

```
aws lakeformation get-data-lake-settings \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "CatalogId": "123456789111"  
}
```

Output:



```
{
  "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"
    ]
  }
}
```

Per ulteriori informazioni, consulta [Modifica delle impostazioni di sicurezza predefinite per il tuo data lake nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [GetDataLakeSettings AWS CLI Command Reference](#).

## get-effective-permissions-for-path

Il seguente esempio di codice mostra come utilizzare `get-effective-permissions-for-path`.

### AWS CLI

Per recuperare le autorizzazioni sulle risorse situate in un percorso specifico

L'`get-effective-permissions-for-path` seguente restituisce i permessi di Lake Formation per una tabella o una risorsa di database specificata situata in un percorso in Amazon S3.

```
aws lakeformation get-effective-permissions-for-path \  
--cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "CatalogId": "123456789111",  
  "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111"  
}
```

Output:

```
{  
  "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":
"E5S1JDSTZ1eUp6SWpvaU9UQTN0RE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjbUYwYVc5dUlqcDdJbk5sWTI5dVpITW1P
}

```

Per ulteriori informazioni, consulta [Managing Lake Formation permissions](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consulta [GetEffectivePermissionsForPath AWS CLI Command Reference](#).

## get-lf-tag

Il seguente esempio di codice mostra come utilizzare `get-lf-tag`.

### AWS CLI

Per recuperare la definizione del tag LF

L'`get-lf-tag` esempio seguente recupera la definizione del tag LF.

```

aws lakeformation get-lf-tag \
  --catalog-id '123456789111' \
  --tag-key 'usergroup'

```

### Output:

```

{
  "CatalogId": "123456789111",
  "TagKey": "usergroup",
  "TagValues": [

```

```
    "analyst",  
    "campaign",  
    "developer"  
  ]  
}
```

Per ulteriori informazioni, consulta [Managing LF-Tags for metadata access control](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consultate Command [GetLfTagReference](#) AWS CLI .

## get-query-state

Il seguente esempio di codice mostra come utilizzare `get-query-state`.

### AWS CLI

Per recuperare lo stato di una richiesta inviata

L'`get-query-state` esempio seguente restituisce lo stato di una query inviata in precedenza.

```
aws lakeformation get-query-state \  
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

Output:

```
{  
  "State": "FINISHED"  
}
```

Per ulteriori informazioni, consulta [Transactional data operations](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetQueryStateReference](#).

## get-query-statistics

Il seguente esempio di codice mostra come utilizzare `get-query-statistics`.

### AWS CLI

Per recuperare le statistiche delle interrogazioni

L'`get-query-statistics` esempio seguente recupera le statistiche sulla pianificazione e l'esecuzione di un'interrogazione.

```
aws lakeformation get-query-statistics \  
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Transactional data operations](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetQueryStatistics](#) Reference.

## get-resource-lf-tags

Il seguente esempio di codice mostra come utilizzare `get-resource-lf-tags`.

AWS CLI

Per elencare i tag LF

L'`list-lf-tags` esempio seguente restituisce un elenco di tag LF che il richiedente è autorizzato a visualizzare.

```
aws lakeformation list-lf-tags \  
  --cli-input-json file://input.json
```

## Contenuto di input.json.

```
{
  "CatalogId": "123456789111",
  "ResourceShareType": "ALL",
  "MaxResults": 2
}
```

## Output:

```
{
  "LFTags": [{
    "CatalogId": "123456789111",
    "TagKey": "category",
    "TagValues": [
      "private",
      "public"
    ]
  },
  {
    "CatalogId": "123456789111",
    "TagKey": "group",
    "TagValues": [
      "analyst",
      "campaign",
      "developer"
    ]
  }
],
  "NextToken": "kIiwiZXhwaXJhdGlvbiI6eyJzZWNVbmRzIjoxNjYwMDY4dCI6ZmFsc2V9"
}
```

Per ulteriori informazioni, consulta [Managing LF-Tags for metadata access control](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consultate Command [GetResourceLfTags](#)Reference AWS CLI .

## get-table-objects

Il seguente esempio di codice mostra come utilizzare `get-table-objects`.

### AWS CLI

Per elencare gli oggetti della tabella governata

L'get-table-objects seguente restituisce l'insieme di oggetti Amazon S3 che costituiscono la tabella governata specificata.

```
aws lakeformation get-table-objects \  
  --cli-input-json file://input.json
```

Contenuto di input.json.

```
{  
  "CatalogId": "012345678901",  
  "DatabaseName": "tpc",  
  "TableName": "dl_tpc_household_demographics_gov",  
  "QueryAsOfTime": "2022-08-10T15:00:00"  
}
```

Output:

```
{  
  "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  
    }]  
  }]  
}
```

Per ulteriori informazioni, consulta [Leggere e scrivere nel data lake all'interno delle transazioni nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [GetTableObjects AWS CLI Command Reference](#).

## get-work-unit-results

Il seguente esempio di codice mostra come utilizzare get-work-unit-results.

### AWS CLI

Per recuperare le unità di lavoro di una determinata query



L'`get-work-unit-results` seguente restituisce le unità di lavoro risultanti dalla query.

```
aws lakeformation get-work-units \
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b' \
  --work-unit-id '0' \
  --work-unit-token 'B2fMSdmQXe9umX8Ux8XCo4=' outfile
```

Output:

```
outfile with Blob content.
```

Per ulteriori informazioni, consulta [Transactional data operations](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetWorkUnitResults](#) Reference.

## get-work-units

Il seguente esempio di codice mostra come utilizzare `get-work-units`.

### AWS CLI

Per recuperare le unità di lavoro

L'`get-work-units` seguente recupera le unità di lavoro generate dall'`StartQueryPlanning` operazione.

```
aws lakeformation get-work-units \
  --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

Output:

```
{
  "WorkUnitRanges": [{
    "WorkUnitIdMax": 0,
    "WorkUnitIdMin": 0,
    "WorkUnitToken":
    "1234eMAk4kL04umqEL4Z5WuxL04AXwABABVhd3MtY3J5cHRvLXB1YmxpYy1rZXkAREEwYm9QbkhINmFYTWphbmMxZW
+f88jzGrYq22gE6jkQlp0B
+0et2eqNumFudAAAAfjB8BqkqhkiG9w0BBwagbzBtAgEAMGgGCSqGSIB3DQEHATAeBg1ghkgBZQMEAS4wEQQMCOEWRda
wAAAAEAAAAAAAAAAAAAAAAEAAACX3/w5h75QAPomfKH+cyEKYU1yccUmB1
+VSojiG0tdsUk7vcjYXUUboYm3dvqRqX2s4gROM0n
```

```
+Ij8R0/8jYmnHkpvYAFNVRPyETyIKg7k5Z9+5I1c2d3446Jw/moWGGxjH8AEG9h27ytm0hozxD0Ei/
F2ZoXz6wLGDfGUo/2WxCkY0hTyNaw6TM
+7drtM7yrW4iNVLUM0LX0xnFjIAhLhooWJek6vjQZUAZzB1AjBH8okRtYP8R7AY2W1s/
hqFBhG0V4l42AC0LxsuZbMQrE2SszWUZ0E9Uew7/n0cyX4CMQDR79INyv4ysMByW9kKGGKyba+cCNk1ExMR
+btBQBmMuB2fMSdmQXe9umX8Ux8XCo4="
  }],
  "QueryId": "1234273f-4a62-4cda-8d98-69615ee8be9b"
}
```

Per ulteriori informazioni, consulta [Transactional data operations](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetWorkUnits](#) Reference.

## grant-permissions

Il seguente esempio di codice mostra come utilizzare `grant-permissions`.

### AWS CLI

Esempio 1: concedere autorizzazioni al principale sulle risorse utilizzando i tag LF

L'`grant-permissions` seguente concede le ALL autorizzazioni al principale su una risorsa di database che corrisponde alla politica LF-Tag.

```
aws lakeformation grant-permissions \
  --cli-input-json file://input.json
```

Contenuto di `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"
]
}

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Concessione e revoca delle autorizzazioni sulle risorse del Data Catalog](#) nella AWS Lake Formation Developer Guide.

Esempio 2: concedere autorizzazioni a livello di colonna al principale

L'grant-permissioneseempio seguente concede il permesso di selezionare una colonna specifica al principale.

```

aws lakeformation grant-permissions \
  --cli-input-json file://input.json

```

Contenuto di `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": []  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Concessione e revoca delle autorizzazioni sulle risorse del Data Catalog](#) nella AWS Lake Formation Developer Guide.

Esempio 3: concedere i permessi di tabella al principale

L'`grant-permissionese`sempio seguente concede al principale il permesso di selezione su tutte le tabelle di un determinato database.

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "CatalogId": "123456789111",  
  "Principal": {  
    "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"  
  },  
  "Resource": {  
    "Table": {  
      "CatalogId": "123456789111",  
      "DatabaseName": "tpc",  
      "TableWildcard": {}  
    }  
  },  
  "Permissions": [  
    "SELECT"  
  ],  
  "PermissionsWithGrantOption": []  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Concessione e revoca delle autorizzazioni sulle risorse del Data Catalog](#) nella AWS Lake Formation Developer Guide.

#### Esempio 4: concedere le autorizzazioni sui tag LF al principale

L'`grant-permission`esempio seguente concede al principale l'autorizzazione di associazione sui tag LF.

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

Contenuto di `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": []  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Concessione e revoca delle autorizzazioni sulle risorse del Data Catalog](#) nella AWS Lake Formation Developer Guide.

#### Esempio 5: concedere le autorizzazioni sulle posizioni dei dati al responsabile

L'`grant-permission`esempio seguente concede al responsabile l'autorizzazione sulla localizzazione dei dati.

```
aws lakeformation grant-permissions \  
  --cli-input-json file://input.json
```

```
--cli-input-json file://input.json
```

Contenuto di `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": []
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Concessione e revoca delle autorizzazioni sulle risorse del Data Catalog](#) nella AWS Lake Formation Developer Guide.

- Per i API dettagli, consulta [GrantPermissions](#) Command Reference.AWS CLI

## **list-data-cells-filter**

Il seguente esempio di codice mostra come utilizzare `list-data-cells-filter`.

AWS CLI

Per elencare i filtri delle celle di dati

L'`list-data-cells-filter` esempio seguente elenca il filtro delle celle di dati per una determinata tabella.

```
aws lakeformation list-data-cells-filter \  
  --cli-input-json file://input.json
```

## Contenuto di input.json.

```
{
  "MaxResults": 2,
  "Table": {
    "CatalogId": "123456789111",
    "DatabaseName": "tpc",
    "Name": "dl_tpc_promotion"
  }
}
```

## Output:

```
{
  "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"
}

```

Per ulteriori informazioni, consulta [Filtraggio dei dati e sicurezza a livello di cella in Lake Formation nella Lake Formation Developer AWS Guide](#).

- Per API i dettagli, consulta Command [ListDataCellsFilter](#) Reference AWS CLI .

## list-permissions

Il seguente esempio di codice mostra come utilizzare `list-permissions`.

### AWS CLI

Esempio 1: per recuperare l'elenco delle autorizzazioni principali sulla risorsa

L'`list-permissions` seguente restituisce un elenco delle autorizzazioni principali sulle risorse del database.

```
aws lakeformation list-permissions \
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{
  "CatalogId": "123456789111",
  "ResourceType": "DATABASE",
  "MaxResults": 2
}
```

Output:

```
{
  "PrincipalResourcePermissions": [{
    "Principal": {
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
campaign-manager"
    }
  }
]
```



```

    },
    "Resource": {
      "Database": {
        "CatalogId": "123456789111",
        "Name": "tpc"
      }
    },
    "Permissions": [
      "DESCRIBE"
    ],
    "PermissionsWithGrantOption": []
  ]],
  "NextToken":
  "E5S1JDSTZ1eUp6SWpvaU9UQTN0RE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjbUYwYVc5dUlqcDdJbk5sWTI5dVpITW1P
}

```

Per ulteriori informazioni, consulta [Managing Lake Formation permissions](#) nella AWS Lake Formation Developer Guide.

Esempio 2: per recuperare l'elenco delle autorizzazioni principali nella tabella con filtri di dati

L'`list-permissions` seguente elenca le autorizzazioni sulla tabella con i relativi filtri di dati concessi al principale.

```

aws lakeformation list-permissions \
  --cli-input-json file://input.json

```

Contenuto di `input.json`.

```

{
  "CatalogId": "123456789111",
  "Resource": {
    "Table": {
      "CatalogId": "123456789111",
      "DatabaseName": "tpc",
      "Name": "dl_tpc_customer"
    }
  },
  "IncludeRelated": "TRUE",
  "MaxResults": 10
}

```

## Output:

```
{
  "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"
}

```

Per ulteriori informazioni, consulta [Managing Lake Formation permissions](#) nella AWS Lake Formation Developer Guide.

Esempio 3: Per recuperare l'elenco delle autorizzazioni principali sui tag LF

L'`list-permissions` seguente elenca le autorizzazioni sui tag LF concesse al principale.

```

aws lakeformation list-permissions \
  --cli-input-json file://input.json

```

## Contenuto di input.json.

```
{
  "CatalogId": "123456789111",
  "Resource": {
    "LFTag": {
      "CatalogId": "123456789111",
      "TagKey": "category",
      "TagValues": [
        "private"
      ]
    }
  },
  "MaxResults": 10
}
```

## Output:

```
{
  "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": "EJwY21GMGFYX0XVJanA3SW50cm1pc3Npb25zIjpmYWxzZX0="
}

```

Per ulteriori informazioni, consulta [Managing Lake Formation permissions](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consulta [ListPermissions AWS CLI Command Reference](#).

## list-resources

Il seguente esempio di codice mostra come utilizzare `list-resources`.

### AWS CLI

Per elencare le risorse gestite dalla Lake Formation

L'`list-resources` esempio seguente elenca le risorse che corrispondono alla condizione gestita da Lake Formation.

```

aws lakeformation list-resources \
  --cli-input-json file://input.json

```

Contenuto di `input.json`.

```
{
  "FilterConditionList": [{
    "Field": "ROLE_ARN",
    "ComparisonOperator": "CONTAINS",
    "StringValueList": [
      "123456789111"
    ]
  }],
  "MaxResults": 10
}
```

Output:

```
{
  "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/EMRLFS3Role",
    "LastModified": "2022-07-29T16:22:03.211000+00:00"
  }
]
}
```

Per ulteriori informazioni, consulta [Managing Lake Formation permissions](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consulta [ListResources AWS CLI Command Reference](#).

## list-transactions

Il seguente esempio di codice mostra come utilizzare `list-transactions`.

### AWS CLI

Per elencare tutti i dettagli delle transazioni

L'`list-transaction`esempio seguente restituisce i metadati relativi alle transazioni e al loro stato.

```
aws lakeformation list-transactions \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "CatalogId": "123456789111",  
  "StatusFilter": "ALL",  
  "MaxResults": 3  
}
```

Output:

```
{  
  "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+X21hHsZLGNC DK3nNGpwRdFpicS0HgcX1/  
QMoniUAKcpR3kj3ts3PVdMA=="  
}
```

Per ulteriori informazioni, consulta [Leggere e scrivere nel data lake all'interno delle transazioni nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [ListTransactions AWS CLI Command Reference](#).

## put-data-lake-settings

Il seguente esempio di codice mostra come utilizzare `put-data-lake-settings`.

### AWS CLI

Per configurare le impostazioni AWS del data lake gestito da Lake Formation

L'esempio seguente imposta l'elenco degli amministratori del data lake e altre impostazioni del data lake.

```
aws lakeformation put-data-lake-settings \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "DataLakeSettings": {  
    "DataLakeAdmins": [{  
      "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-  
admin"  
    }  
  ],  
  "CreateDatabaseDefaultPermissions": [],  
  "CreateTableDefaultPermissions": [],  
  "TrustedResourceOwners": [],  
  "AllowExternalDataFiltering": true,  
  "ExternalDataFilteringAllowList": [{  
    "DataLakePrincipalIdentifier": "123456789111"  
  }],  
  "AuthorizedSessionTagValueList": ["Amazon EMR"]  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Modifica delle impostazioni di sicurezza predefinite per il tuo data lake nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [PutDataLakeSettings AWS CLI Command Reference](#).



## register-resource

Il seguente esempio di codice mostra come utilizzare `register-resource`.

### AWS CLI

Esempio 1: per registrare lo storage del data lake utilizzando Service Linked Role

L'esempio seguente registra la risorsa come gestita da Lake Formation utilizzando il ruolo collegato al servizio.

```
aws lakeformation register-resource \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",  
  "UseServiceLinkedRole": true  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere una posizione Amazon S3 al data lake nella AWS Lake Formation Developer Guide](#).

Esempio 2: per registrare lo storage del data lake utilizzando un ruolo personalizzato

L'esempio seguente registra la risorsa come gestita da Lake Formation utilizzando un ruolo personalizzato.

```
aws lakeformation register-resource \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",  
  "UseServiceLinkedRole": false,  
  "RoleArn": "arn:aws:iam::123456789111:role/LF-GlueServiceRole"  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere una posizione Amazon S3 al data lake nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [RegisterResource AWS CLI Command Reference](#).

## **remove-lf-tags-from-resource**

Il seguente esempio di codice mostra come utilizzare `remove-lf-tags-from-resource`.

### AWS CLI

Per rimuovere il tag LF da una risorsa

L'esempio seguente rimuove l'associazione LF-tag con la risorsa della tabella.

```
aws lakeformation remove-lf-tags-from-resource \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "CatalogId": "123456789111",  
  "Resource": {  
    "Table": {  
      "CatalogId": "123456789111",  
      "DatabaseName": "tpc",  
      "Name": "dl_tpc_promotion"  
    }  
  },  
  "LFTags": [{  
    "CatalogId": "123456789111",  
    "TagKey": "usergroup",  
    "TagValues": [  
      "developer"  
    ]  
  }]  
}
```

Output:

```
{
  "Failures": []
}
```

Per ulteriori informazioni, consulta [Assegnazione di tag LF alle risorse del Data Catalog](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, vedere [RemoveLfTagsFromResource](#) in AWS CLI Command Reference.

## revoke-permissions

Il seguente esempio di codice mostra come utilizzare `revoke-permissions`.

### AWS CLI

Per revocare le autorizzazioni sulle risorse al principale

L'esempio seguente revoca l'accesso principale a una tabella specifica di un determinato database.

```
aws lakeformation revoke-permissions \
  --cli-input-json file://input.json
```

Contenuto di `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": []
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Concessione e revoca delle autorizzazioni sulle risorse del Data Catalog](#) nella AWS Lake Formation Developer Guide.

- Per i API dettagli, consulta [RevokePermissions](#) Command Reference.AWS CLI

## search-databases-by-lf-tags

Il seguente esempio di codice mostra come utilizzare `search-databases-by-lf-tags`.

### AWS CLI

Per cercare nelle risorse del database per LFTags

L'`search-databases-by-lf-tags` esempio seguente cerca le risorse del database che corrispondono all'LFTagespressione.

```
aws lakeformation search-databases-by-lf-tags \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "MaxResults": 1,  
  "CatalogId": "123456789111",  
  "Expression": [{  
    "TagKey": "usergroup",  
    "TagValues": [  
      "developer"  
    ]  
  }]  
}
```

Output:

```
{  
  "DatabaseList": [{  
    "Database": {  
      "CatalogId": "123456789111",  
      "Name": "tpc"  
    },  
  },  
}
```

```
    "LFTags": [{
      "CatalogId": "123456789111",
      "TagKey": "usergroup",
      "TagValues": [
        "developer"
      ]
    }]
  ]
}
```

Per ulteriori informazioni, consulta [Visualizzazione delle risorse a cui è assegnato un tag LF](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [SearchDatabasesByLfTags](#) Reference.

## search-tables-by-lf-tags

Il seguente esempio di codice mostra come utilizzare `search-tables-by-lf-tags`.

### AWS CLI

Per cercare nelle risorse della tabella per LFTags

L'esempio seguente esegue una ricerca sulle risorse della tabella che corrispondono all'espressione LFTags.

```
aws lakeformation search-tables-by-lf-tags \
  --cli-input-json file:///input.json
```

Contenuto di `input.json`.

```
{
  "MaxResults": 2,
  "CatalogId": "123456789111",
  "Expression": [{
    "TagKey": "usergroup",
    "TagValues": [
      "developer"
    ]
  }]
}
```

## Output:

```
{
  "NextToken": "c2VhcmNoQWxsVGFnc0luVGFibGVzIjpmYWxzZX0=",
  "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"
      ]
    }]
  }
]
}]
}

```

Per ulteriori informazioni, consulta [Visualizzazione delle risorse a cui è assegnato un tag LF](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [SearchTablesByLfTagsReference](#).

## start-query-planning

Il seguente esempio di codice mostra come utilizzare `start-query-planning`.

### AWS CLI

Per elaborare l'istruzione di interrogazione

L'`start-query-planning`esempio seguente invia una richiesta per l'elaborazione di un'istruzione di query.

```
aws lakeformation start-query-planning \  
  --cli-input-json file://input.json
```

Contenuto di `input.json`.

```
{  
  "QueryPlanningContext": {  
    "CatalogId": "012345678901",  
    "DatabaseName": "tpc"  
  },  
  "QueryString": "select * from dl_tpc_household_demographics_gov where  
hd_income_band_sk=9"  
}
```

Output:

```
{  
  "QueryId": "772a273f-4a62-4cda-8d98-69615ee8be9b"  
}
```

Per ulteriori informazioni, consulta [Leggere e scrivere nel data lake all'interno delle transazioni nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [StartQueryPlanning AWS CLI Command Reference](#).

## start-transaction

Il seguente esempio di codice mostra come utilizzare `start-transaction`.

### AWS CLI

Per iniziare una nuova transazione

L'`start-transaction` esempio seguente avvia una nuova transazione e ne restituisce l'ID.

```
aws lakeformation start-transaction \  
  --transaction-type = 'READ_AND_WRITE'
```

Output:

```
{
```

```
"TransactionId": "b014d972ca8347b89825e33c5774aec4"  
}
```

Per ulteriori informazioni, consulta [Leggere e scrivere nel data lake all'interno delle transazioni nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [StartTransaction AWS CLI Command Reference](#).

## update-lf-tag

Il seguente esempio di codice mostra come utilizzare `update-lf-tag`.

### AWS CLI

Per aggiornare la definizione del tag LF

L'`update-lf-tag` esempio seguente aggiorna la definizione del tag LF.

```
aws lakeformation update-lf-tag \  
  --catalog-id '123456789111' \  
  --tag-key 'usergroup' \  
  --tag-values-to-add ['admin']
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing LF-Tags for metadata access control](#) nella AWS Lake Formation Developer Guide.

- Per API i dettagli, consultate Command [UpdateLfTag Reference AWS CLI](#) .

## update-table-objects

Il seguente esempio di codice mostra come utilizzare `update-table-objects`.

### AWS CLI

Per modificare gli oggetti della tabella governata

L'`update-table-object` esempio seguente aggiunge gli oggetti S3 forniti alla tabella governata specificata.

```
aws lakeformation update-table-objects \  
  --table-name 'table-name' \  
  --s3-objects 's3-objects'
```

```
--cli-input-json file://input.json
```

Contenuto di 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
    }
  ]
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Leggere e scrivere nel data lake all'interno delle transazioni nella AWS Lake Formation Developer Guide](#).

- Per API i dettagli, consulta [UpdateTableObjects AWS CLI Command Reference](#).

## Esempi di utilizzo di Lambda AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Lambda.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **add-layer-version-permission**

Il seguente esempio di codice mostra come utilizzare `add-layer-version-permission`.

#### AWS CLI

Per aggiungere autorizzazioni a una versione di livello

L'esempio seguente concede all'account specificato l'autorizzazione a utilizzare la versione 1 del layer `my-layer`

```
aws lambda add-layer-version-permission \
  --layer-name my-layer \
  --statement-id xaccount \
  --action lambda:GetLayerVersion \
  --principal 123456789012 \
  --version-number 1
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [AWS Lambda Layers nella Lambda Developer AWS Guide](#).

- Per API i dettagli, consulta AWS CLI Command [AddLayerVersionPermission](#) Reference.

### **add-permission**

Il seguente esempio di codice mostra come utilizzare `add-permission`.

## AWS CLI

Per aggiungere autorizzazioni a una funzione Lambda esistente

L'add-permissionesempio seguente concede al SNS servizio Amazon l'autorizzazione a richiamare una funzione denominata. my-function

```
aws lambda add-permission \  
  --function-name my-function \  
  --action lambda:InvokeFunction \  
  --statement-id sns \  
  --principal sns.amazonaws.com
```

Output:

```
{  
  "Statement":  
  {  
    "Sid": "sns",  
    "Effect": "Allow",  
    "Principal": {  
      "Service": "sns.amazonaws.com"  
    },  
    "Action": "lambda:InvokeFunction",  
    "Resource": "arn:aws:lambda:us-east-2:123456789012:function:my-function"  
  }  
}
```

Per ulteriori informazioni, consulta [Using Resource-based Policies for Lambda nella AWS Lambda Developer Guide](#).AWS

- Per i API dettagli, consulta Command Reference. [AddPermission](#)AWS CLI

## create-alias

Il seguente esempio di codice mostra come utilizzarecreate-alias.

## AWS CLI

Per creare un alias per una funzione Lambda

L'create-aliasesempio seguente crea un alias denominato LIVE che punta alla versione 1 della funzione my-function Lambda.



```
aws lambda create-alias \  
  --function-name my-function \  
  --description "alias for live version of function" \  
  --function-version 1 \  
  --name LIVE
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Configurazione degli alias delle funzioni AWS Lambda](#) nella AWS Lambda Developer Guide.

- Per i API dettagli, consulta [CreateAlias](#) Command Reference.AWS CLI

## create-event-source-mapping

Il seguente esempio di codice mostra come utilizzare `create-event-source-mapping`.

AWS CLI

Per creare una mappatura tra un'origine di eventi e una funzione AWS Lambda

L'`create-event-source-mapping` esempio seguente crea una mappatura tra una SQS coda e la funzione `my-function`.

```
aws lambda create-event-source-mapping \  
  --function-name my-function \  
  --batch-size 5 \  
  --event-source-arn arn:aws:sqs:us-west-2:123456789012:mySQSqueue
```

Output:

```
{  
  "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"
}
```

Per ulteriori informazioni, consulta [AWS Lambda Event Source Mapping](#) nella Lambda AWS Developer Guide.

- Per API i dettagli, consulta Command [CreateEventSourceMapping](#) Reference AWS CLI .

## create-function

Il seguente esempio di codice mostra come utilizzare `create-function`.

### AWS CLI

Per creare una funzione Lambda

L'esempio di `create-function` seguente crea una funzione Lambda denominata `my-function`.

```
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
```

Contenuto di `my-function.zip`.

```
This file is a deployment package that contains your function code and any dependencies.
```

Output:

```
{
  "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": ""
}
```

Per ulteriori informazioni, consulta [Configurazione della funzione Lambda AWS](#) nella Guida per gli sviluppatori di AWS .

- Per API i dettagli, vedere [CreateFunction](#) in AWS CLI Command Reference.

## delete-alias

Il seguente esempio di codice mostra come utilizzare `delete-alias`.

### AWS CLI

Per eliminare un alias di una funzione Lambda

L'`delete-alias` esempio seguente elimina l'alias denominato `LIVE` dalla funzione Lambda `my-function`.

```
aws lambda delete-alias \  
  --function-name my-function \  
  --name LIVE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configurazione degli alias delle funzioni AWS Lambda](#) nella AWS Lambda Developer Guide.

- Per i API dettagli, consulta [DeleteAlias](#) Command Reference. AWS CLI

## delete-event-source-mapping

Il seguente esempio di codice mostra come utilizzare `delete-event-source-mapping`.

### AWS CLI

Per eliminare la mappatura tra un'origine di eventi e una funzione AWS Lambda

L'esempio seguente elimina la mappatura tra una SQS coda e la funzione `my-function` Lambda.

```
aws lambda delete-event-source-mapping \  
  --uuid a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [AWS Lambda Event Source Mapping](#) nella Lambda AWS Developer Guide.

- Per API i dettagli, consulta Command [DeleteEventSourceMapping](#) Reference AWS CLI .

## delete-function-concurrency

Il seguente esempio di codice mostra come utilizzare `delete-function-concurrency`.

### AWS CLI

Per rimuovere il limite di esecuzione simultanea riservato da una funzione

L'esempio seguente elimina il limite di esecuzione simultanea riservato dalla funzione `my-function`

```
aws lambda delete-function-concurrency \  
  --function-name my-function
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Reserved Concurrency for a Lambda Function nella Lambda Developer Guide](#).AWS

- Per i API dettagli, consulta [DeleteFunctionConcurrency](#)Command Reference.AWS CLI

## **delete-function-event-invoke-config**

Il seguente esempio di codice mostra come utilizzare `delete-function-event-invoke-config`.

AWS CLI

Per eliminare una configurazione di chiamata asincrona

L'esempio seguente elimina la configurazione di chiamata asincrona per l'alias della funzione specificata. GREEN

```
aws lambda delete-function-event-invoke-config --function-name my-function:GREEN
```

- Per API i dettagli, vedere in Command Reference. [DeleteFunctionEventInvokeConfig](#)AWS CLI

## **delete-function**

Il seguente esempio di codice mostra come utilizzare `delete-function`.

AWS CLI

Esempio 1: eliminazione di una funzione Lambda in base al nome della funzione

L'esempio di `delete-function` seguente elimina la funzione Lambda denominata `my-function` specificandone il nome.

```
aws lambda delete-function \  
  --function-name my-function
```

Questo comando non produce alcun output.

## Esempio 2: eliminare una funzione Lambda per funzione ARN

L'`delete-function` esempio seguente elimina la funzione Lambda `my-function` denominata specificando quella della funzione. ARN

```
aws lambda delete-function \  
  --function-name arn:aws:lambda:us-west-2:123456789012:function:my-function
```

Questo comando non produce alcun output.

## Esempio 3: eliminare una funzione Lambda tramite una funzione parziale ARN

L'`delete-function` esempio seguente elimina la funzione Lambda `my-function` denominata specificando il parziale della funzione. ARN

```
aws lambda delete-function \  
  --function-name 123456789012:function:my-function
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Configurazione delle funzioni Lambda AWS](#) nella Guida per gli sviluppatori di AWS .

- Per API i dettagli, vedere [DeleteFunction](#) in AWS CLI Command Reference.

## `delete-layer-version`

Il seguente esempio di codice mostra come utilizzare `delete-layer-version`.

### AWS CLI

Per eliminare una versione di un layer Lambda

L'`delete-layer-version` esempio seguente elimina la versione 2 del layer denominato. `my-layer`

```
aws lambda delete-layer-version \  
  --layer-name my-layer \  
  --version-number 2
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [AWS Lambda Layers nella Lambda Developer AWS Guide](#).

- Per API i dettagli, consulta AWS CLI Command [DeleteLayerVersionReference](#).

## delete-provisioned-concurrency-config

Il seguente esempio di codice mostra come utilizzare `delete-provisioned-concurrency-config`.

### AWS CLI

Per eliminare una configurazione di concorrenza assegnata

L'esempio seguente elimina la configurazione di concorrenza fornita per l'alias `GREEN` della funzione specificata.

```
aws lambda delete-provisioned-concurrency-config \  
  --function-name my-function \  
  --qualifier GREEN
```

- Per i API dettagli, vedere [DeleteProvisionedConcurrencyConfig](#) in Command Reference.AWS CLI

## get-account-settings

Il seguente esempio di codice mostra come utilizzare `get-account-settings`.

### AWS CLI

Per recuperare i dettagli sul tuo account in una regione AWS

L'esempio seguente mostra i limiti Lambda e le informazioni sull'utilizzo per il tuo account.

```
aws lambda get-account-settings
```

Output:

```
{  
  "AccountLimit": {  
    "CodeSizeUnzipped": 262144000,  
    "UnreservedConcurrentExecutions": 1000,
```

```
    "ConcurrentExecutions": 1000,
    "CodeSizeZipped": 52428800,
    "TotalCodeSize": 80530636800
  },
  "AccountUsage": {
    "FunctionCount": 4,
    "TotalCodeSize": 9426
  }
}
```

Per ulteriori informazioni, consulta [AWS Lambda Limits nella Lambda Developer AWS Guide](#).

- Per API i dettagli, consulta AWS CLI Command [GetAccountSettings](#)Reference.

## get-alias

Il seguente esempio di codice mostra come utilizzare `get-alias`.

### AWS CLI

Per recuperare dettagli su un alias di funzione

L'`get-alias`esempio seguente visualizza i dettagli per l'alias denominato nella LIVE funzione `my-function` Lambda.

```
aws lambda get-alias \
  --function-name my-function \
  --name LIVE
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Configurazione degli alias delle funzioni AWS Lambda](#) nella AWS Lambda Developer Guide.

- Per i API dettagli, consulta [GetAlias](#)Command Reference.AWS CLI



## get-event-source-mapping

Il seguente esempio di codice mostra come utilizzare `get-event-source-mapping`.

### AWS CLI

Per recuperare i dettagli sulla mappatura della sorgente di un evento

L'esempio seguente mostra i dettagli per la mappatura tra una SQS coda e la funzione Lambda `my-function`.

```
aws lambda get-event-source-mapping \
  --uuid "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [AWS Lambda Event Source Mapping](#) nella Lambda AWS Developer Guide.

- Per API i dettagli, consulta Command [GetEventSourceMapping](#) Reference AWS CLI .

## get-function-concurrency

Il seguente esempio di codice mostra come utilizzare `get-function-concurrency`.

### AWS CLI

Per visualizzare l'impostazione della concorrenza riservata per una funzione

L'esempio seguente recupera l'impostazione della concorrenza riservata per la funzione specificata.

```
aws lambda get-function-concurrency \  
  --function-name my-function
```

Output:

```
{  
  "ReservedConcurrentExecutions": 250  
}
```

- Per API i dettagli, vedere [GetFunctionConcurrency](#) in AWS CLI Command Reference.

## get-function-configuration

Il seguente esempio di codice mostra come utilizzare `get-function-configuration`.

AWS CLI

Per recuperare le impostazioni specifiche della versione di una funzione Lambda

L'`get-function-configuration` esempio seguente visualizza le impostazioni per la versione 2 della funzione. `my-function`

```
aws lambda get-function-configuration \  
  --function-name my-function:2
```

Output:

```
{  
  "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-uy3l9qqq",  
  "Timeout": 3,  
  "Runtime": "nodejs10.x",  
  "TracingConfig": {  
    "Mode": "PassThrough"  
  },  
  "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJmIKidWoaCgk=",  
  "Description": "",
```

```
"VpcConfig": {
  "SubnetIds": [],
  "VpcId": "",
  "SecurityGroupIds": []
},
"CodeSize": 304,
"FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:2",
"Handler": "index.handler"
}
```

Per ulteriori informazioni, consulta [Configurazione della funzione Lambda AWS](#) nella Guida per gli sviluppatori di AWS .

- Per API i dettagli, vedere [GetFunctionConfiguration](#) in AWS CLI Command Reference.

## get-function-event-invoke-config

Il seguente esempio di codice mostra come utilizzare `get-function-event-invoke-config`.

### AWS CLI

Per visualizzare una configurazione di chiamata asincrona

L'`get-function-event-invoke-config` seguente recupera la configurazione di chiamata asincrona per l'alias della funzione specificata. `BLUE`

```
aws lambda get-function-event-invoke-config \
  --function-name my-function:BLUE
```

Output:

```
{
  "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"
    }
  }
}
```

```
}  
}
```

- Per API i dettagli, vedere in Command Reference. [GetFunctionEventInvokeConfig](#) AWS CLI

## get-function

Il seguente esempio di codice mostra come utilizzare `get-function`.

### AWS CLI

Per recuperare le informazioni relative a una funzione

Nell'esempio di `get-function` seguente vengono visualizzate informazioni sulla funzione `my-function`.

```
aws lambda get-function \  
  --function-name my-function
```

Output:

```
{  
  "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-uy3l9qq",
    "Timeout": 3,
    "LastModified": "2019-09-24T18:20:35.054+0000",
    "Runtime": "nodejs10.x",
    "Description": ""
  }
}
```

Per ulteriori informazioni, consulta [Configurazione della funzione Lambda AWS](#) nella Guida per gli sviluppatori di AWS .

- Per API i dettagli, vedere [GetFunction](#) in AWS CLI Command Reference.

## get-layer-version-by-arn

Il seguente esempio di codice mostra come utilizzare `get-layer-version-by-arn`.

### AWS CLI

Per recuperare informazioni su una versione del layer Lambda

L'`get-layer-version-by-arn` esempio seguente mostra informazioni sulla versione del layer con l'Amazon Resource Name specificato (ARN).

```
aws lambda get-layer-version-by-arn \
  --arn "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-SciPy1x:2"
```

Output:

```
{
  "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": "GGmv8ocUw4c1y0T8HL0Vx/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"
  }

```

Per ulteriori informazioni, consulta [AWS Lambda Layers nella Lambda Developer AWS Guide](#).

- Per API i dettagli, consulta AWS CLI Command [GetLayerVersionByArnReference](#).

## get-layer-version-policy

Il seguente esempio di codice mostra come utilizzare `get-layer-version-policy`.

### AWS CLI

Per recuperare la politica di autorizzazione per una versione del layer Lambda

L'esempio seguente visualizza le informazioni sulle politiche relative alla versione 1 per il layer denominato `my-layer`

```

aws lambda get-layer-version-policy \
  --layer-name my-layer \
  --version-number 1

```

Output:

```

{
  "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"
}
```

Per ulteriori informazioni, consulta [AWS Lambda Layers nella Lambda Developer AWS Guide](#).

- Per API i dettagli, consulta AWS CLI Command [GetLayerVersionPolicy](#) Reference.

## get-layer-version

Il seguente esempio di codice mostra come utilizzare `get-layer-version`.

### AWS CLI

Per recuperare informazioni su una versione del layer Lambda

L'`get-layer-version` esempio seguente visualizza le informazioni per la versione 1 del layer denominato `my-layer`

```
aws lambda get-layer-version \
  --layer-name my-layer \
  --version-number 1
```

Output:

```
{
  "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"
]
}
```

Per ulteriori informazioni, consulta [AWS Lambda Layers nella Lambda Developer AWS Guide](#).

- Per API i dettagli, consulta AWS CLI Command [GetLayerVersion](#) Reference.

## get-policy

Il seguente esempio di codice mostra come utilizzare `get-policy`.

### AWS CLI

Per recuperare la IAM politica basata sulle risorse per una funzione, una versione o un alias

L'output seguente visualizza le informazioni sulle politiche relative alla funzione `my-function` Lambda.

```
aws lambda get-policy \
  --function-name my-function
```

Output:

```
{
  "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"
}

```

Per ulteriori informazioni, consulta [Using Resource-based Policies for Lambda nella AWS Lambda Developer Guide](#).AWS

- Per i API dettagli, consulta Command Reference. [GetPolicy](#)AWS CLI

## get-provisioned-concurrency-config

Il seguente esempio di codice mostra come utilizzare `get-provisioned-concurrency-config`.

### AWS CLI

Per visualizzare una configurazione di concorrenza assegnata

L'`get-provisioned-concurrency-config`esempio seguente visualizza i dettagli della configurazione di concorrenza fornita per l'`BLUE`alias della funzione specificata.

```

aws lambda get-provisioned-concurrency-config \
  --function-name my-function \
  --qualifier BLUE

```

Output:

```

{
  "RequestedProvisionedConcurrentExecutions": 100,
  "AvailableProvisionedConcurrentExecutions": 100,
  "AllocatedProvisionedConcurrentExecutions": 100,
  "Status": "READY",
  "LastModified": "2019-12-31T20:28:49+0000"
}

```

- Per API i dettagli, vedere [GetProvisionedConcurrencyConfig](#)in AWS CLI Command Reference.

## invoke

Il seguente esempio di codice mostra come utilizzare `invoke`.

## AWS CLI

### Esempio 1: come richiamare una funzione Lambda in modo sincrono

L'esempio di `invoke` seguente richiama la funzione `my-function` in modo sincrono. L'opzione `cli-binary-format` è obbligatoria se si utilizza la AWS CLI versione 2. Per ulteriori informazioni, consulta le [opzioni della riga di comando globali AWS CLI supportate](#) nella Guida per l'utente dell'interfaccia a riga di AWS comando.

```
aws lambda invoke \  
  --function-name my-function \  
  --cli-binary-format raw-in-base64-out \  
  --payload '{ "name": "Bob" }' \  
  response.json
```

Output:

```
{  
  "ExecutedVersion": "$LATEST",  
  "StatusCode": 200  
}
```

Per ulteriori informazioni, consulta [Chiamata sincrona](#) nella Guida per gli sviluppatori di AWS Lambda.

### Esempio 2: come richiamare una funzione Lambda in modo asincrono

L'esempio di `invoke` seguente richiama la funzione `my-function` in modo asincrono. L'opzione `cli-binary-format` è obbligatoria se si utilizza la AWS CLI versione 2. Per ulteriori informazioni, consulta le [opzioni della riga di comando globali AWS CLI supportate](#) nella Guida per l'utente dell'interfaccia a riga di AWS comando.

```
aws lambda invoke \  
  --function-name my-function \  
  --invocation-type Event \  
  --cli-binary-format raw-in-base64-out \  
  --payload '{ "name": "Bob" }' \  
  response.json
```

Output:

```
{
  "StatusCode": 202
}
```

Per ulteriori informazioni, consulta [Chiamata asincrona](#) nella Guida per gli sviluppatori di AWS Lambda.

- Per API i dettagli, consulta [Invoke](#) in AWS CLI Command Reference.

## list-aliases

Il seguente esempio di codice mostra come utilizzare `list-aliases`.

### AWS CLI

Per recuperare l'elenco degli alias per una funzione Lambda

L'esempio seguente visualizza un elenco degli alias per la funzione `my-function` Lambda.

```
aws lambda list-aliases \
  --function-name my-function
```

Output:

```
{
  "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"
    }
  ]
}
```

```
    }
  ]
}
```

Per ulteriori informazioni, consulta [Configurazione degli alias delle funzioni AWS Lambda](#) nella AWS Lambda Developer Guide.

- Per i API dettagli, consulta [ListAliases](#) Command Reference.AWS CLI

## list-event-source-mappings

Il seguente esempio di codice mostra come utilizzare `list-event-source-mappings`.

### AWS CLI

Per elencare le mappature delle sorgenti degli eventi per una funzione

L'`list-event-source-mappings` esempio seguente visualizza un elenco delle mappature delle sorgenti degli eventi per la funzione `my-function`.

```
aws lambda list-event-source-mappings \
  --function-name my-function
```

### Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [AWS Lambda Event Source Mapping](#) nella Lambda AWS Developer Guide.

- Per API i dettagli, consulta Command [ListEventSourceMappingsReference](#) AWS CLI .

## list-function-event-invoke-configs

Il seguente esempio di codice mostra come utilizzare `list-function-event-invoke-configs`.

### AWS CLI

Per visualizzare un elenco di configurazioni di chiamata asincrona

L'`list-function-event-invoke-configs`esempio seguente elenca le configurazioni di chiamata asincrona per la funzione specificata.

```
aws lambda list-function-event-invoke-configs \  
  --function-name my-function
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

- Per i API dettagli, vedere in Command Reference. [ListFunctionEventInvokeConfigs](#)AWS CLI

## list-functions

Il seguente esempio di codice mostra come utilizzare `list-functions`.

## AWS CLI

Come recuperare un elenco di funzioni Lambda

L'esempio di `list-functions` seguente visualizza un elenco di tutte le funzioni per l'utente attuale.

```
aws lambda list-functions
```

Output:

```
{
  "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/1TxCuQqK3gDZP3i8gUoqUUVrM6E=",
      "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": ""
    }
]
}
```

Per ulteriori informazioni, consulta [Configurazione della funzione Lambda AWS](#) nella Guida per gli sviluppatori di AWS .

- Per API i dettagli, vedere [ListFunctions](#) in AWS CLI Command Reference.

## list-layer-versions

Il seguente esempio di codice mostra come utilizzare `list-layer-versions`.

### AWS CLI

Per elencare le versioni di un layer AWS Lambda

L'`list-layers-versions` esempio seguente visualizza informazioni sulle versioni del layer denominato `my-layer`.

```
aws lambda list-layer-versions \
  --layer-name my-layer
```

Output:

```
{
  "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"
      ]
    }
  ]
}
```



```
    ]
  }
]
}
```

Per ulteriori informazioni, consulta [AWS Lambda Layers nella Lambda Developer AWS Guide](#).

- Per API i dettagli, consulta AWS CLI Command [ListLayerVersions](#)Reference.

## list-layers

Il seguente esempio di codice mostra come utilizzare `list-layers`.

### AWS CLI

Per elencare i livelli compatibili con il runtime della funzione

L'esempio seguente mostra informazioni sui livelli compatibili con il runtime di Python 3.11.

```
aws lambda list-layers \
  --compatible-runtime python3.11
```

Output:

```
{
  "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"
        ]
      }
    }
  ]
}
```

```
}
```

Per ulteriori informazioni, consulta [AWS Lambda Layers nella Lambda Developer AWS Guide](#).

- Per API i dettagli, consulta AWS CLI Command [ListLayers](#) Reference.

## list-provisioned-concurrency-configs

Il seguente esempio di codice mostra come utilizzare `list-provisioned-concurrency-configs`.

### AWS CLI

Per ottenere un elenco delle configurazioni di concorrenza assegnate

L'`list-provisioned-concurrency-configs` seguente elenca le configurazioni di concorrenza assegnate per la funzione specificata.

```
aws lambda list-provisioned-concurrency-configs \  
  --function-name my-function
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

```
]
}
```

- Per API i dettagli, vedere [ListProvisionedConcurrencyConfigs](#) in AWS CLI Command Reference.

## list-tags

Il seguente esempio di codice mostra come utilizzare `list-tags`.

### AWS CLI

Per recuperare l'elenco dei tag per una funzione Lambda

L'`list-tags` esempio seguente visualizza i tag associati alla funzione `my-function` Lambda.

```
aws lambda list-tags \
  --resource arn:aws:lambda:us-west-2:123456789012:function:my-function
```

Output:

```
{
  "Tags": {
    "Category": "Web Tools",
    "Department": "Sales"
  }
}
```

Per ulteriori informazioni, consulta [Tagging Lambda Functions nella Lambda Developer AWS Guide](#).

- Per API i dettagli, consulta Command [ListTags](#) Reference AWS CLI .

## list-versions-by-function

Il seguente esempio di codice mostra come utilizzare `list-versions-by-function`.

### AWS CLI

Per recuperare un elenco di versioni di una funzione

L'`list-versions-by-function` esempio seguente visualizza l'elenco delle versioni della funzione `my-function` Lambda.

```
aws lambda list-versions-by-function \  
--function-name my-function
```

Output:

```
{  
  "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-uy3l9qyq",
    "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-uy3l9qyq",
    "Timeout": 3,
    "LastModified": "2019-10-01T16:47:28.490+0000",
    "Runtime": "nodejs10.x",
    "Description": "newer version"
  }
]
}
```

Per ulteriori informazioni, consulta [Configurazione degli alias delle funzioni AWS Lambda](#) nella AWS Lambda Developer Guide.

- Per i API dettagli, consulta [ListVersionsByFunction](#) Command Reference.AWS CLI

## publish-layer-version

Il seguente esempio di codice mostra come utilizzare `publish-layer-version`.

### AWS CLI

Per creare una versione del layer Lambda

L'esempio seguente crea una nuova versione del livello di libreria Python. Il comando recupera il contenuto del livello in un file denominato `layer.zip` nel bucket S3 specificato.

```
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
```

Output:

```
{  
  "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"  
  ]  
}
```

```
]
}
```

Per ulteriori informazioni, consulta [AWS Lambda Layers nella Lambda Developer AWS Guide](#).

- Per API i dettagli, consulta AWS CLI Command [PublishLayerVersionReference](#).

## publish-version

Il seguente esempio di codice mostra come utilizzare `publish-version`.

### AWS CLI

Per pubblicare una nuova versione di una funzione

L'esempio seguente pubblica una nuova versione della funzione `my-function` Lambda.

```
aws lambda publish-version \
  --function-name my-function
```

Output:

```
{
  "TracingConfig": {
    "Mode": "PassThrough"
  },
  "CodeSha256": "dBG9m8SGdm1Ejw/JYX1hhvCrAv5TxvXsbl/RMr0fT/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": ""
}
```

Per ulteriori informazioni, consulta [Configurazione degli alias delle funzioni AWS Lambda](#) nella AWS Lambda Developer Guide.

- Per i API dettagli, consulta [PublishVersion](#) Command Reference.AWS CLI

## **put-function-concurrency**

Il seguente esempio di codice mostra come utilizzare `put-function-concurrency`.

### AWS CLI

Per configurare un limite di concorrenza riservato per una funzione

L'`put-function-concurrency` esempio seguente configura 100 esecuzioni simultanee riservate per la funzione. `my-function`

```
aws lambda put-function-concurrency \
  --function-name my-function \
  --reserved-concurrent-executions 100
```

Output:

```
{
  "ReservedConcurrentExecutions": 100
}
```

Per ulteriori informazioni, consulta [Reserved Concurrency for a Lambda Function nella Lambda Developer](#) Guide.AWS

- Per i API dettagli, consulta [PutFunctionConcurrency](#) Command Reference.AWS CLI

## **put-function-event-invoke-config**

Il seguente esempio di codice mostra come utilizzare `put-function-event-invoke-config`.

### AWS CLI

Per configurare la gestione degli errori per la chiamata asincrona

L'`put-function-event-invoke-config` esempio seguente imposta la durata massima degli eventi di un'ora e disabilita i nuovi tentativi per la funzione specificata.



```
aws lambda put-function-event-invoke-config \  
  --function-name my-function \  
  --maximum-event-age-in-seconds 3600 \  
  --maximum-retry-attempts 0
```

Output:

```
{  
  "LastModified": 1573686021.479,  
  "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-function:  
$LATEST",  
  "MaximumRetryAttempts": 0,  
  "MaximumEventAgeInSeconds": 3600,  
  "DestinationConfig": {  
    "OnSuccess": {},  
    "OnFailure": {}  
  }  
}
```

- Per API i dettagli, vedere [PutFunctionEventInvokeConfig](#) in AWS CLI Command Reference.

## put-provisioned-concurrency-config

Il seguente esempio di codice mostra come utilizzare `put-provisioned-concurrency-config`.

AWS CLI

Per allocare la concorrenza assegnata

L'esempio seguente alloca 100 valute simultanee assegnate per l'alias della BLUE funzione specificata.

```
aws lambda put-provisioned-concurrency-config \  
  --function-name my-function \  
  --qualifier BLUE \  
  --provisioned-concurrent-executions 100
```

Output:

```
{  
  "Requested ProvisionedConcurrentExecutions": 100,
```

```
"Allocated ProvisionedConcurrentExecutions": 0,  
"Status": "IN_PROGRESS",  
"LastModified": "2019-11-21T19:32:12+0000"  
}
```

- Per i API dettagli, vedere [PutProvisionedConcurrencyConfig](#) in Command Reference.AWS CLI

## **remove-layer-version-permission**

Il seguente esempio di codice mostra come utilizzare `remove-layer-version-permission`.

### AWS CLI

Per eliminare le autorizzazioni relative alla versione di livello

L'esempio seguente elimina l'autorizzazione di un account a configurare una versione di livello.

```
aws lambda remove-layer-version-permission \  
  --layer-name my-layer \  
  --statement-id xaccount \  
  --version-number 1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [AWS Lambda Layers nella Lambda Developer AWS Guide](#).

- Per API i dettagli, consulta AWS CLI Command [RemoveLayerVersionPermission](#) Reference.

## **remove-permission**

Il seguente esempio di codice mostra come utilizzare `remove-permission`.

### AWS CLI

Per rimuovere le autorizzazioni da una funzione Lambda esistente

L'esempio seguente rimuove l'autorizzazione a richiamare una funzione denominata `my-function`

```
aws lambda remove-permission \  
  --function-name my-function \  
  --statement-id xaccount
```

```
--function-name my-function \  
--statement-id sns
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Using Resource-based Policies for Lambda nella AWS Lambda Developer Guide](#).AWS

- Per i API dettagli, consulta Command Reference. [RemovePermission](#)AWS CLI

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

AWS CLI

Per aggiungere tag a una funzione Lambda esistente

L'`tag-resource`esempio seguente aggiunge un tag con il nome della chiave DEPARTMENT e un valore di Department A alla funzione Lambda specificata.

```
aws lambda tag-resource \  
--resource arn:aws:lambda:us-west-2:123456789012:function:my-function \  
--tags "DEPARTMENT=Department A"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Lambda Functions nella Lambda Developer AWS Guide](#).

- Per API i dettagli, consulta Command [TagResource](#)Reference AWS CLI .

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

AWS CLI

Per rimuovere i tag da una funzione Lambda esistente

L'`untag-resource`esempio seguente rimuove il tag con il DEPARTMENT tag key name dalla funzione `my-function` Lambda.

```
aws lambda untag-resource \  
  --resource arn:aws:lambda:us-west-2:123456789012:function:my-function \  
  --tag-keys DEPARTMENT
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Lambda Functions nella Lambda Developer AWS Guide](#).

- Per API i dettagli, consulta Command [UntagResource](#)Reference AWS CLI .

## update-alias

Il seguente esempio di codice mostra come utilizzare `update-alias`.

### AWS CLI

Per aggiornare l'alias di una funzione

L'`update-alias`esempio seguente aggiorna l'alias denominato in modo LIVE che punti alla versione 3 della funzione `my-function` Lambda.

```
aws lambda update-alias \  
  --function-name my-function \  
  --function-version 3 \  
  --name LIVE
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Configurazione degli alias delle funzioni AWS Lambda](#) nella AWS Lambda Developer Guide.

- Per i API dettagli, consulta [UpdateAlias](#)Command Reference.AWS CLI

## update-event-source-mapping

Il seguente esempio di codice mostra come utilizzare `update-event-source-mapping`.

### AWS CLI

Per aggiornare la mappatura tra un'origine di eventi e una funzione AWS Lambda

L'`update-event-source-mapping`esempio seguente aggiorna la dimensione del batch a 8 nella mappatura specificata.

```
aws lambda update-event-source-mapping \  
  --uuid "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \  
  --batch-size 8
```

Output:

```
{  
  "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:mysqlqueue"  
}
```

Per ulteriori informazioni, consulta [AWS Lambda Event Source Mapping](#) nella Lambda AWS Developer Guide.

- Per API i dettagli, consulta Command [UpdateEventSourceMapping](#)Reference AWS CLI .

## update-function-code

Il seguente esempio di codice mostra come utilizzare `update-function-code`.

### AWS CLI

Come aggiornare il codice di una funzione Lambda

L'`update-function-code`esempio seguente sostituisce il codice della versione non pubblicata (\$LATEST) della `my-function` funzione con il contenuto del file zip specificato.

```
aws lambda update-function-code \  
  --function-name my-function \  
  --zip-file fileb://my-function.zip
```

Output:

```
{  
  "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-uy319qqq",  
  "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"  
}
```

Per ulteriori informazioni, consulta [Configurazione della funzione Lambda AWS](#) nella Guida per gli sviluppatori di AWS .

- Per API i dettagli, vedere [UpdateFunctionCode](#) in AWS CLI Command Reference.

## update-function-configuration

Il seguente esempio di codice mostra come utilizzare `update-function-configuration`.

### AWS CLI

Come modificare la configurazione di una funzione

L'update-function-configurationesempio seguente modifica la dimensione della memoria in 256 MB per la versione non pubblicata (\$LATEST) della my-function funzione.

```
aws lambda update-function-configuration \  
  --function-name my-function \  
  --memory-size 256
```

Output:

```
{  
  "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-uy319yyq",  
  "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"  
}
```

Per ulteriori informazioni, consulta [Configurazione della funzione Lambda AWS](#) nella Guida per gli sviluppatori di AWS .

- Per API i dettagli, vedere [UpdateFunctionConfiguration](#) in AWS CLI Command Reference.

## update-function-event-invoke-config

Il seguente esempio di codice mostra come utilizzareupdate-function-event-invoke-config.

## AWS CLI

Per aggiornare una configurazione di chiamata asincrona

L'update-function-event-invoke-configesempio seguente aggiunge una destinazione in caso di errore alla configurazione di chiamata asincrona esistente per la funzione specificata.

```
aws lambda update-function-event-invoke-config \  
  --function-name my-function \  
  --destination-config '{"OnFailure":{"Destination": "arn:aws:sqs:us-  
east-2:123456789012:destination"}}'
```

Output:

```
{  
  "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"  
    }  
  }  
}
```

- Per i API dettagli, vedere in Command Reference. [UpdateFunctionEventInvokeConfigAWS CLI](#)

## Esempi di License Manager che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with License Manager.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.



## Argomenti

- [Azioni](#)

## Azioni

### **create-license-configuration**

Il seguente esempio di codice mostra come utilizzare `create-license-configuration`.

#### AWS CLI

Esempio 1: Per creare una configurazione di licenza

L'esempio seguente crea una configurazione di licenza con un limite rigido di 10 core.

```
aws license-manager create-license-configuration --name my-license-configuration \  
  --license-counting-type Core \  
  --license-count 10 \  
  --license-count-hard-limit
```

Output:

```
{  
  "LicenseConfigurationArn": "arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE1111"  
}
```

Esempio 2: creare una configurazione di licenza

L'esempio seguente crea una configurazione di licenza con un limite minimo di 100 vCPUs. Utilizza una regola per abilitare CPU l'ottimizzazione v.

```
aws license-manager create-license-configuration --name my-license-configuration \  
  --license-counting-type vCPU \  
  --license-count 100 \  
  --license-rules "#honorVcpuOptimization=true"
```

Output:

```
{
  "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE2222"
}
```

- Per API i dettagli, vedere [CreateLicenseConfiguration](#) in AWS CLI Command Reference.

## delete-license-configuration

Il seguente esempio di codice mostra come utilizzare `delete-license-configuration`.

### AWS CLI

Per eliminare una configurazione di licenza

L'`delete-license-configuration` esempio seguente elimina la configurazione di licenza specificata.

```
aws license-manager delete-license-configuration \
  --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteLicenseConfiguration](#) in AWS CLI Command Reference.

## get-license-configuration

Il seguente esempio di codice mostra come utilizzare `get-license-configuration`.

### AWS CLI

Per ottenere informazioni sulla configurazione della licenza

L'`get-license-configuration` esempio seguente visualizza i dettagli per la configurazione della licenza specificata.

```
aws license-manager get-license-configuration \
  --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

## Output:

```
{
  "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
    }
  ]
}
```

```
]
}
```

- Per API i dettagli, vedere [GetLicenseConfiguration](#) in AWS CLI Command Reference.

## get-service-settings

Il seguente esempio di codice mostra come utilizzare `get-service-settings`.

### AWS CLI

Per ottenere le impostazioni del License Manager

L'`get-service-settings` esempio seguente visualizza le impostazioni del servizio per License Manager nella regione corrente.

```
aws license-manager get-service-settings
```

Di seguito viene illustrato un esempio di output se l'individuazione di risorse tra account è disabilitata.

```
{
  "OrganizationConfiguration": {
    "EnableIntegration": false
  },
  "EnableCrossAccountsDiscovery": false
}
```

Di seguito viene mostrato un esempio di output se il rilevamento di risorse tra account è abilitato.

```
{
  "S3BucketArn": "arn:aws:s3:::aws-license-manager-service-c22d6279-35c4-47c4-bb",
  "OrganizationConfiguration": {
    "EnableIntegration": true
  },
  "EnableCrossAccountsDiscovery": true
}
```

- Per API i dettagli, vedere [GetServiceSettings](#) in AWS CLI Command Reference.

## list-associations-for-license-configuration

Il seguente esempio di codice mostra come utilizzare `list-associations-for-license-configuration`.

### AWS CLI

Per ottenere le associazioni per una configurazione di licenza

L'`list-associations-for-license-configuration`esempio seguente visualizza informazioni dettagliate per le associazioni della configurazione di licenza specificata.

```
aws license-manager list-associations-for-license-configuration \
  --license-configuration-arn arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

Output:

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [ListAssociationsForLicenseConfiguration](#) in AWS CLI Command Reference.

## list-license-configurations

Il seguente esempio di codice mostra come utilizzare `list-license-configurations`.

## AWS CLI

Esempio 1: per elencare tutte le configurazioni di licenza

L'`list-license-configurations` seguente elenca tutte le configurazioni di licenza.

```
aws license-manager list-license-configurations
```

Output:

```
{
  "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
    }
  ]
},
{
  ...
}
]
}

```

Esempio 2: Per elencare una configurazione di licenza specifica

L'`list-license-configurations` esempio seguente elenca solo la configurazione di licenza specificata.

```

aws license-manager list-license-configurations \
  --license-configuration-arns arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE

```

- Per API i dettagli, vedere [ListLicenseConfigurations](#) in AWS CLI Command Reference.

## `list-license-specifications-for-resource`

Il seguente esempio di codice mostra come utilizzare `list-license-specifications-for-resource`.

### AWS CLI

Per elencare le configurazioni di licenza per una risorsa

L'`list-license-specifications-for-resource` esempio seguente elenca le configurazioni di licenza associate all'Amazon Machine Image (AMI) specificata.

```
aws license-manager list-license-specifications-for-resource \  
--resource-arn arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0
```

Output:

```
{  
  "LicenseConfigurationArn": "arn:aws:license-manager:us-  
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE"  
}
```

- Per API i dettagli, consulta [ListLicenseSpecificationsForResource AWS CLI Command Reference](#).

## list-resource-inventory

Il seguente esempio di codice mostra come utilizzare `list-resource-inventory`.

### AWS CLI

Per elencare le risorse nell'inventario delle risorse

L'`list-resource-inventory` seguente elenca le risorse gestite utilizzando l'inventario di Systems Manager.

```
aws license-manager list-resource-inventory
```

Output:

```
{  
  "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"
    }
]
}

```

- Per API i dettagli, vedere [ListResourceInventory](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag per una configurazione di licenza

L'`list-tags-for-resource` esempio seguente elenca i tag per la configurazione di licenza specificata.

```

aws license-manager list-tags-for-resource \
  --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE

```

Output:

```

{
  "Tags": [
    {
      "Key": "project",

```

```

    "Value": "lima"
  }
]
}

```

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## list-usage-for-license-configuration

Il seguente esempio di codice mostra come utilizzare `list-usage-for-license-configuration`.

### AWS CLI

Per elencare le licenze in uso per una configurazione di licenza

L'`list-usage-for-license-configuration` esempio seguente elenca le informazioni sulle risorse che utilizzano le licenze per la configurazione di licenza specificata. Ad esempio, se il tipo di licenza è vCPU, ogni istanza consuma una licenza per v. CPU

```

aws license-manager list-usage-for-license-configuration \
  --license-configuration-arn arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE

```

Output:

```

{
  "LicenseConfigurationUsageList": [
    {
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/i-04a636d18e83cfac",
      "ResourceType": "EC2_INSTANCE",
      "ResourceStatus": "running",
      "ResourceOwnerId": "123456789012",
      "AssociationTime": 1570892850.519,
      "ConsumedLicenses": 2
    }
  ]
}

```

- Per API i dettagli, vedere [ListUsageForLicenseConfiguration](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere un tag, una configurazione di licenza

L'`tag-resource` esempio seguente aggiunge il tag specificato (nome e valore della chiave) alla configurazione di licenza specificata.

```
aws license-manager tag-resource \  
  --tags Key=project,Value=Lima \  
  --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una configurazione di licenza

L'`untag-resource` esempio seguente rimuove il tag specificato (nome chiave e risorsa) dalla configurazione di licenza specificata.

```
aws license-manager untag-resource \  
  --tag-keys project \  
  --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-license-configuration

Il seguente esempio di codice mostra come utilizzare `update-license-configuration`.

## AWS CLI

Per aggiornare la configurazione di una licenza

L'`update-license-configuration` seguente aggiorna la configurazione di licenza specificata per rimuovere il limite rigido.

```
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
```

Questo comando non produce alcun output.

L'`update-license-configuration` seguente aggiorna la configurazione di licenza specificata per modificarne lo stato in `DISABLED`.

```
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
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [UpdateLicenseConfiguration](#) in AWS CLI Command Reference.

## update-license-specifications-for-resource

Il seguente esempio di codice mostra come utilizzare `update-license-specifications-for-resource`.

## AWS CLI

Per aggiornare le configurazioni di licenza per una risorsa

L'`update-license-specifications-for-resource` seguente sostituisce la configurazione della licenza associata all'Amazon Machine Image (AMI) specificata rimuovendo una configurazione di licenza e aggiungendone un'altra.

```
aws license-manager update-license-specifications-for-resource \  
  --resource-arn arn:aws:ec2:us-west-2:image/ami-1234567890abcdef0 \  
  --license-specification-arn arn:aws:license-manager:us-west-2:880185128111:license-specification:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

```
--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
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [UpdateLicenseSpecificationsForResource AWS CLI Command Reference](#).

## update-service-settings

Il seguente esempio di codice mostra come utilizzare `update-service-settings`.

### AWS CLI

Per aggiornare le impostazioni del License Manager

L'esempio seguente abilita l'individuazione delle risorse tra account per License Manager nella AWS regione corrente. Il bucket Amazon S3 è il Resource Data Sync necessario per l'inventario di Systems Manager.

```
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
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta AWS CLI Command [UpdateServiceSettings](#) Reference.

## Esempi di utilizzo di Lightsail AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface con Lightsail.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **allocate-static-ip**

Il seguente esempio di codice mostra come utilizzare `allocate-static-ip`.

#### AWS CLI

Per creare un IP statico

L'esempio seguente crea l'IP statico specificato, che può essere collegato a un'istanza.

```
aws lightsail allocate-static-ip \  
  --static-ip-name StaticIp-1
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

```
}
```

- Per API i dettagli, vedere [AllocateStaticIp](#) in AWS CLI Command Reference.

## attach-disk

Il seguente esempio di codice mostra come utilizzare `attach-disk`.

### AWS CLI

Per collegare un disco di archiviazione a blocchi a un'istanza

L'esempio seguente collega il disco `Disk-1` all'istanza `WordPress_Multisite-1` con il percorso del disco di `/dev/xvdf`

```
aws lightsail attach-disk \  
  --disk-name Disk-1 \  
  --disk-path /dev/xvdf \  
  --instance-name WordPress_Multisite-1
```

Output:

```
{  
  "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
  }
]
}

```

- Per API i dettagli, vedere [AttachDisk](#) in AWS CLI Command Reference.

## attach-instances-to-load-balancer

Il seguente esempio di codice mostra come utilizzare `attach-instances-to-load-balancer`.

### AWS CLI

Per collegare istanze a un sistema di bilanciamento del carico

L'esempio seguente collega le istanze e MEAN-3 al sistema di bilanciamento del carico MEAN-1 MEAN-2. `LoadBalancer-1`

```

aws lightsail attach-instances-to-load-balancer \
  --instance-names {"MEAN-1","MEAN-2","MEAN-3"} \
  --load-balancer-name LoadBalancer-1

```

Output:

```

{
  "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
  }
]
}

```

- Per API i dettagli, vedere [AttachInstancesToLoadBalancer](#) in AWS CLI Command Reference.

## attach-load-balancer-tls-certificate

Il seguente esempio di codice mostra come utilizzare `attach-load-balancer-tls-certificate`.

### AWS CLI

Per allegare un TLS certificato a un sistema di bilanciamento del carico

L'esempio seguente allega il TLS certificato di bilanciamento del carico `Certificate2` al sistema di bilanciamento del carico.

`LoadBalancer-1`

```
aws lightsail attach-load-balancer-tls-certificate \  
  --certificate-name Certificate2 \  
  --load-balancer-name LoadBalancer-1
```

Output:

```
{  
  "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
    }
  ]
}

```

- Per API i dettagli, vedere [AttachLoadBalancerTlsCertificate](#) in AWS CLI Command Reference.

## attach-static-ip

Il seguente esempio di codice mostra come utilizzare `attach-static-ip`.

### AWS CLI

Per collegare un IP statico a un'istanza

L'`attach-static-ip` esempio seguente collega un IP statico `StaticIp-1` all'istanza `MEAN-1`.

```

aws lightsail attach-static-ip \
  --static-ip-name StaticIp-1 \
  --instance-name MEAN-1

```

Output:

```

{
  "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
    }
]
}

```

- Per API i dettagli, vedere [AttachStaticIp](#) in AWS CLI Command Reference.

## close-instance-public-ports

Il seguente esempio di codice mostra come utilizzare `close-instance-public-ports`.

### AWS CLI

Per chiudere le porte del firewall per un'istanza

L'`close-instance-public-ports` esempio seguente chiude la TCP porta 22 sull'istanza `MEAN-2`.

```

aws lightsail close-instance-public-ports \
  --instance-name MEAN-2 \
  --port-info fromPort=22,protocol=TCP,toPort=22

```

Output:

```

{
  "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
  }
}

```

- Per API i dettagli, vedere [CloseInstancePublicPorts](#) in AWS CLI Command Reference.

## copy-snapshot

Il seguente esempio di codice mostra come utilizzare `copy-snapshot`.

### AWS CLI

Esempio 1: copiare un'istantanea all'interno della stessa regione AWS

L'`copy-snapshot` esempio seguente copia un'istantanea dell'istanza `MEAN-1-1571075291` come istantanea dell'istanza `MEAN-1-Copy` all'interno della stessa regione. AWS `us-west-2`

```

aws lightsail copy-snapshot \
  --source-snapshot-name MEAN-1-1571075291 \
  --target-snapshot-name MEAN-1-Copy \
  --source-region us-west-2

```

Output:

```

{
  "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
  }
]
}

```

Per ulteriori informazioni, [consulta Copiare gli snapshot da una AWS regione all'altra in Amazon Lightsail nella Lightsail Dev Guide](#).

Esempio 2: copiare uno snapshot da una regione all'altra AWS

L'copy-snapshotesempio seguente copia lo snapshot dell'istanza MEAN-1-1571075291 come istantanea dell'istanza MEAN-1-1571075291-Copy da AWS Region a. us-west-2 us-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

```

Output:

```

{
  "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
    }
]
}

```

Per ulteriori informazioni, [consulta Copiare gli snapshot da una AWS regione all'altra in Amazon Lightsail nella Lightsail Dev Guide](#).

Esempio 3: copiare uno snapshot automatico all'interno della stessa regione AWS

L'copy-snapshotesempio seguente copia un'istantanea automatica 2019-10-14 dell'istanza WordPress-1 come istantanea manuale WordPress-1-10142019 nella regione. AWS 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

```

Output:

```

{
  "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
    }
  ]
}

```



```
}
```

Per ulteriori informazioni, consulta [Conservazione di istantanee automatiche di istanze o dischi in Amazon Lightsail nella Lightsail Dev Guide](#).

Esempio 4: copiare uno snapshot automatico da una regione all'altra AWS

L'copy-snapshotesempio seguente copia un'istanza automatica 2019-10-14 dell'istanza WordPress-1 come istanza manuale WordPress-1-10142019 dalla AWS regione a. us-west-2 us-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
```

Output:

```
{  
  "operations": [  
    {  
      "id": "dfffa128b-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  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Conservazione di istantanee automatiche di istanze o dischi in Amazon Lightsail nella Lightsail Dev Guide](#).

- Per API i dettagli, consulta Command Reference. [CopySnapshot](#) AWS CLI

## create-disk-from-snapshot

Il seguente esempio di codice mostra come utilizzare `create-disk-from-snapshot`.

### AWS CLI

Per creare un disco, creare un disco da un'istantanea del disco

L'esempio seguente crea un disco di archiviazione a blocchi denominato `Disk-2` in base allo snapshot del disco di archiviazione a blocchi specificato. Il disco viene creato nella AWS regione e nella zona di disponibilità specificate, con 32 GB di spazio di archiviazione.

```
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
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Creazione di un disco di storage a blocchi da uno snapshot in Amazon Lightsail nella Lightsail Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [CreateDiskFromSnapshot](#) AWS CLI

## create-disk-snapshot

Il seguente esempio di codice mostra come utilizzare `create-disk-snapshot`.

### AWS CLI

Esempio 1: creare un'istantanea di un disco

L'`create-disk-snapshot` seguente crea un'istantanea denominata `DiskSnapshot-1` del disco di archiviazione a blocchi specificato.

```
aws lightsail create-disk-snapshot \  
  --disk-name Disk-1 \  
  --disk-snapshot-name DiskSnapshot-1
```

Output:

```
{  
  "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
  }
]
}

```

Esempio 2: creare un'istantanea del disco di sistema di un'istanza

L'`create-disk-snapshotesempio` seguente crea un'istantanea del disco di sistema dell'istanza specificata.

```

aws lightsail create-disk-snapshot \
  --instance-name WordPress-1 \
  --disk-snapshot-name SystemDiskSnapshot-1

```

Output:

```

{
  "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
  }
]
}

```

Per ulteriori informazioni, consulta [Snapshot in Amazon Lightsail e Creazione di un'istantanea del volume root di un'istanza in Amazon Lightsail nella Lightsail Developer Guide](#).

- Per i dettagli, consulta Command Reference. API [CreateDiskSnapshot](#)AWS CLI

## create-disk

Il seguente esempio di codice mostra come utilizzare `create-disk`.

### AWS CLI

Per creare un disco di archiviazione a blocchi

L'`create-disk`esempio seguente crea un disco di archiviazione a blocchi `Disk-1` nella AWS regione e nella zona di disponibilità specificate, con 32 GB di spazio di archiviazione.

```

aws lightsail create-disk \
  --disk-name Disk-1 \
  --availability-zone us-west-2a \
  --size-in-gb 32

```

Output:

```

{
  "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
  }
]
}

```

- Per API i dettagli, vedere [CreateDisk](#) in AWS CLI Command Reference.

## create-domain-entry

Il seguente esempio di codice mostra come utilizzare `create-domain-entry`.

### AWS CLI

Per creare una voce di dominio (DNSrecord)

L'esempio seguente crea un DNS record (A) per l'apice del dominio specificato che punta all'indirizzo IP di un'istanza.

Nota: le API operazioni relative al dominio di Lightsail sono disponibili solo nella Regione. `us-east-1` Se il CLI profilo è configurato per utilizzare una regione diversa, è necessario includere il `--region us-east-1` parametro o il comando avrà esito negativo.

```

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

```

Output:

```

{
  "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
  }
}
```

Per ulteriori informazioni, consulta [DNS Amazon Lightsail e Creazione di DNS una zona per la gestione dei record DNS del dominio in Amazon Lightsail nella Lightsail Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [CreateDomainEntry](#) AWS CLI

## create-domain

Il seguente esempio di codice mostra come utilizzare `create-domain`.

### AWS CLI

Per creare un dominio (DNS zona)

L'esempio seguente crea una DNS zona per il dominio specificato.

Nota: le API operazioni relative al dominio di Lightsail sono disponibili solo nella Regione. `us-east-1` Se il CLI profilo è configurato per utilizzare una regione diversa, è necessario includere il `--region us-east-1` parametro o il comando avrà esito negativo.

```
aws lightsail create-domain \
  --region us-east-1 \
  --domain-name example.com
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [DNS Amazon Lightsail e Creazione di DNS una zona per la gestione dei record DNS del dominio in Amazon Lightsail nella Lightsail Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [CreateDomain](#) AWS CLI

## create-instance-snapshot

Il seguente esempio di codice mostra come utilizzare `create-instance-snapshot`.

AWS CLI

Per creare un'istantanea di un'istanza

L'`create-instance-snapshot` seguente crea un'istantanea dall'istanza specificata.

```
aws lightsail create-instance-snapshot \
  --instance-name WordPress-1 \
  --instance-snapshot-name WordPress-Snapshot-1
```

Output:

```
{
  "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
}
]
}

```

- Per API i dettagli, vedere [CreateInstanceSnapshot](#) in AWS CLI Command Reference.

## create-instances-from-snapshot

Il seguente esempio di codice mostra come utilizzare `create-instances-from-snapshot`.

### AWS CLI

Per creare un'istanza da un'istantanea

L'esempio seguente crea un'istanza dallo snapshot dell'istanza specificata, nella AWS regione e nella zona di disponibilità specificate, utilizzando il pacchetto USD \$12.

Nota: il pacchetto specificato deve avere specifiche uguali o superiori a quelle dell'istanza di origine originale utilizzata per creare l'istantanea.

```
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
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

- Per API i dettagli, vedere [CreateInstancesFromSnapshot](#) in AWS CLI Command Reference.

## create-instances

Il seguente esempio di codice mostra come utilizzare `create-instances`.

### AWS CLI

Esempio 1: per creare una singola istanza

L'`create-instances` esempio seguente crea un'istanza nella AWS regione e nella zona di disponibilità specificate, utilizzando il WordPress blueprint e il pacchetto da \$5,00USD.

```
aws lightsail create-instances \  
  --instance-names Instance-1 \  
  --availability-zone us-west-2a \  
  --bundle-id small_3_0
```

```
--availability-zone us-west-2a \  
--blueprint-id wordpress \  
--bundle-id nano_3_0
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

Esempio 2: per creare più istanze contemporaneamente

L'create-instancesesempio seguente crea tre istanze nella AWS regione e nella zona di disponibilità specificate, utilizzando il WordPress blueprint e il pacchetto da \$5,00. USD

```
aws lightsail create-instances \  
--instance-names {"Instance1","Instance2","Instance3"} \  
--availability-zone us-west-2a \  
--blueprint-id wordpress \  
--bundle-id nano_3_0
```

Output:

```
{  
  "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
  }
]
}
```

- Per i API dettagli, vedere in Command Reference. [CreateInstances](#) AWS CLI

## create-key-pair

Il seguente esempio di codice mostra come utilizzare `create-key-pair`.

### AWS CLI

Per creare una coppia di chiavi

L'esempio seguente crea una key pair che è possibile utilizzare per autenticarsi e connettersi a un'istanza.

```
aws lightsail create-key-pair \
  --key-pair-name MyPersonalKeyPair
```

L'output fornisce il valore base64 della chiave privata che è possibile utilizzare per l'autenticazione nelle istanze che utilizzano la coppia di chiavi creata. Nota: copia e incolla il valore della chiave privata base64 in una posizione sicura perché non puoi recuperarlo in un secondo momento.

```
{
  "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
+y804UyVsjrUqCaMXDhmfXpWuIMPwuXhwcKh7e8hwoTfkiX0E6Ql
+KqF/MiA3w6DCjEqvvdI07SiEZJFsuGNfYDDN3w60Re15MUhmn30Jdn4y/
A7Nwb3IxL4pPvFE4rgFRKU8n1jp9kwRn1VMVB0WuGXk6n+H6M2f1 ",
  "privateKeyBase64": "-----BEGIN RSA PRIVATE KEY-----
EXAMPLETCCAfICCQD6m7oRw0uX0jANBgqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
\nVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6\nnb24xFDASBgNVBAsTC01BTSBD
\nBgqhkiG9w0BCQEWEG5vb251QGFTYXpvi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
\nMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
```

```

\nVQQHEwdTZWF0dGx1MQ8wDQEXAMPLEwZBbWF6b24xFDASBgNVBA5TC01BTSBDb25z
\nb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWxhZAdBgqhkiG9w0BCQEWEG5vb251QGft
\nYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMEXAMPLE4GmWIWJ
\n21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
\nrDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
\nIbb30hjZnczvQAaREXAMPLEMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4\nnnUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
\nFFBjvSfpJIIJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780EXAMPLELvix79LjSTb
\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
  }
}

```

- Per API i dettagli, vedere [CreateKeyPair](#) in AWS CLI Command Reference.

## create-load-balancer-tls-certificate

Il seguente esempio di codice mostra come utilizzare `create-load-balancer-tls-certificate`.

### AWS CLI

Per creare un TLS certificato per un sistema di bilanciamento del carico

L'esempio seguente crea un TLS certificato allegato al sistema di bilanciamento del carico specificato. Il certificato creato si applica ai domini specificati. Nota: è possibile creare solo due certificati per un sistema di bilanciamento del carico.

```

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
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

- Per API i dettagli, vedere [CreateLoadBalancerTlsCertificate](#) in AWS CLI Command Reference.

## create-load-balancer

Il seguente esempio di codice mostra come utilizzare `create-load-balancer`.

### AWS CLI

Per creare un sistema di bilanciamento del carico

L'esempio seguente crea un sistema di bilanciamento del carico con un TLS certificato. Il TLS certificato si applica ai domini specificati e indirizza il traffico verso le istanze sulla porta 80.

```
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
```

Output:

```
{
  "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
}
]
}

```

Per ulteriori informazioni, consulta i sistemi di bilanciamento del [carico di Lightsail nella Lightsail Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [CreateLoadBalancer](#) AWS CLI

## create-relational-database-from-snapshot

Il seguente esempio di codice mostra come utilizzare `create-relational-database-from-snapshot`.

### AWS CLI

Per creare un database gestito da un'istantanea

L'esempio seguente crea un database gestito dall'istantanea specificata nella AWS regione e nella zona di disponibilità specificate,

utilizzando il pacchetto di database USD standard da \$15. Nota: il pacchetto specificato deve avere specifiche uguali o superiori a quelle del pacchetto del database di origine utilizzato per creare l'istantanea.

```
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
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

- Per API i dettagli, vedere [CreateRelationalDatabaseFromSnapshot](#) in AWS CLI Command Reference.

## create-relational-database-snapshot

Il seguente esempio di codice mostra come utilizzare `create-relational-database-snapshot`.

### AWS CLI

Per creare un'istantanea di un database gestito

L'create-relational-database-snapshotesempio seguente crea un'istanza del database gestito specificato.

```
aws lightsail create-relational-database-snapshot \  
  --relational-database-name Database1 \  
  --relational-database-snapshot-name RelationalDatabaseSnapshot1
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

```
}
```

- Per API i dettagli, vedere [CreateRelationalDatabaseSnapshot](#) in AWS CLI Command Reference.

## create-relational-database

Il seguente esempio di codice mostra come utilizzare `create-relational-database`.

### AWS CLI

Per creare un database gestito

L'esempio seguente crea un database gestito nella AWS regione e nella zona di disponibilità specificate, utilizzando il motore di database MySQL 5.6 (`mysql_5_6`) e il pacchetto di database standard da \$15 (`micro_1_0`). USD Il database gestito è precompilato con un nome utente principale e non è accessibile pubblicamente.

```
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
```

Output:

```
{  
  "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
      }
    ]
  }

```

- Per API i dettagli, vedere [CreateRelationalDatabase](#) in AWS CLI Command Reference.

## delete-auto-snapshot

Il seguente esempio di codice mostra come utilizzare `delete-auto-snapshot`.

### AWS CLI

Per eliminare un'istantanea automatica

L'esempio seguente elimina l'istantanea `2019-10-10` automatica dell'istanza `WordPress-1`.

```

aws lightsail delete-auto-snapshot \
  --resource-name WordPress-1 \
  --date 2019-10-10

```

Output:

```

{
  "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"
    }
  ]
}

```

```
}
```

Per ulteriori informazioni, consulta [Eliminazione di istantanee automatiche di istanze o dischi in Amazon Lightsail nella Lightsail Dev Guide](#).

- Per i dettagli, consulta Command Reference. API [DeleteAutoSnapshot](#) AWS CLI

## delete-disk-snapshot

Il seguente esempio di codice mostra come utilizzare `delete-disk-snapshot`.

### AWS CLI

Per eliminare un'istanza di un disco di archiviazione a blocchi

L'esempio seguente elimina l'istanza specificata di un disco di archiviazione a blocchi

```
aws lightsail delete-disk-snapshot \
  --disk-snapshot-name DiskSnapshot-1
```

Output:

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [DeleteDiskSnapshot](#) in AWS CLI Command Reference.

## delete-disk

Il seguente esempio di codice mostra come utilizzare `delete-disk`.

### AWS CLI

Per eliminare un disco di archiviazione a blocchi

L'esempio seguente elimina il disco di archiviazione a blocchi specificato.

```
aws lightsail delete-disk \  
  --disk-name Disk-1
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

- Per API i dettagli, vedere [DeleteDisk](#) in AWS CLI Command Reference.

## delete-domain-entry

Il seguente esempio di codice mostra come utilizzare `delete-domain-entry`.

## AWS CLI

Per eliminare una voce di dominio (DNSrecord)

L'`delete-domain-entry` seguente elimina la voce di dominio specificata da un dominio esistente.

Nota: le API operazioni relative al dominio di Lightsail sono disponibili solo nella Regione. `us-east-1` Se il CLI profilo è configurato per utilizzare una regione diversa, è necessario includere il `--region us-east-1` parametro o il comando avrà esito negativo.

```
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
```

Output:

```
{  
  "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  
  }  
}
```

- Per API i dettagli, vedere [DeleteDomainEntry](#) in AWS CLI Command Reference.

## `delete-domain`

Il seguente esempio di codice mostra come utilizzare `delete-domain`.



## AWS CLI

Per eliminare un dominio (DNSzona)

L'`delete-domain` seguente elimina il dominio specificato e tutte le voci in esso contenute (DNSrecords).

Nota: le API operazioni relative al dominio di Lightsail sono disponibili solo nella Regione. `us-east-1` Se il CLI profilo è configurato per utilizzare una regione diversa, è necessario includere il `--region us-east-1` parametro o il comando avrà esito negativo.

```
aws lightsail delete-domain \  
  --region us-east-1 \  
  --domain-name example.com
```

Output:

```
{  
  "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  
  }  
}
```

- Per API i dettagli, vedere [DeleteDomain](#) in AWS CLI Command Reference.

## **delete-instance-snapshot**

Il seguente esempio di codice mostra come utilizzare `delete-instance-snapshot`.

## AWS CLI

titolo

L'`delete-instance-snapshot` seguente elimina l'istantanea specificata di un'istanza.

```
aws lightsail delete-instance-snapshot \  
  --instance-snapshot-name WordPress-1-Snapshot-1
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

- Per API i dettagli, vedete [DeleteInstanceSnapshot](#) in AWS CLI Command Reference.

## `delete-instance`

Il seguente esempio di codice mostra come utilizzare `delete-instance`.

## AWS CLI

Per eliminare un'istanza

L'`delete-instance` seguente elimina l'istanza specificata.

```
aws lightsail delete-instance \  
  --instance-name WordPress-1
```

Output:

```
{  
  "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
}
]
```

- Per API i dettagli, vedere [DeleteInstance](#) in AWS CLI Command Reference.

## delete-key-pair

Il seguente esempio di codice mostra come utilizzare `delete-key-pair`.

### AWS CLI

Per eliminare una coppia di chiavi

L'`delete-key-pair` esempio seguente elimina la coppia di chiavi specificata.

```
aws lightsail delete-key-pair \
  --key-pair-name MyPersonalKeyPair
```

Output:

```
{
  "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
  }
}
```

```
}  
}
```

- Per API i dettagli, vedere [DeleteKeyPair](#) in AWS CLI Command Reference.

## delete-known-host-keys

Il seguente esempio di codice mostra come utilizzare `delete-known-host-keys`.

### AWS CLI

Per eliminare chiavi host note da un'istanza

L'`delete-known-host-keys` seguente elimina la chiave host nota dall'istanza specificata.

```
aws lightsail delete-known-host-keys \  
  --instance-name Instance-1
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Risoluzione dei problemi di connessione con il RDP client SSH o il browser Amazon Lightsail](#) nella Lightsail Dev Guide.

- Per i API dettagli, consulta Command Reference. [DeleteKnownHostKeys](#) AWS CLI

## delete-load-balancer-tls-certificate

Il seguente esempio di codice mostra come utilizzare `delete-load-balancer-tls-certificate`.

### AWS CLI

Per eliminare un TLS certificato per un sistema di bilanciamento del carico

L'esempio seguente elimina il TLS certificato specificato dal sistema di bilanciamento del carico specificato.

```
aws lightsail delete-load-balancer-tls-certificate \  
  --load-balancer-name MyFirstLoadBalancer \  
  --certificate-name MyFirstCertificate
```

Output:

```
{  
  "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
}
]
}

```

- Per API i dettagli, vedere [DeleteLoadBalancerTlsCertificate](#) in AWS CLI Command Reference.

## delete-load-balancer

Il seguente esempio di codice mostra come utilizzare `delete-load-balancer`.

### AWS CLI

Per eliminare un sistema di bilanciamento del carico

L'`delete-load-balancer` esempio seguente elimina il sistema di bilanciamento del carico specificato e tutti i certificati associati. TLS

```

aws lightsail delete-load-balancer \
  --load-balancer-name MyFirstLoadBalancer

```

Output:

```

{
  "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
  }
]
}
```

Per ulteriori informazioni, consulta il titolo nella guida.

- Per API i dettagli, vedere [DeleteLoadBalancer](#) in AWS CLI Command Reference.

## **delete-relational-database-snapshot**

Il seguente esempio di codice mostra come utilizzare `delete-relational-database-snapshot`.



## AWS CLI

Per eliminare un'istanza di un database gestito

L'`delete-relational-database-snapshotesempio` seguente elimina lo snapshot specificato di un database gestito.

```
aws lightsail delete-relational-database-snapshot \  
--relational-database-snapshot-name Database-Oregon-1-1566839359
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

- Per API i dettagli, vedere [DeleteRelationalDatabaseSnapshot](#) in AWS CLI Command Reference.

## `delete-relational-database`

Il seguente esempio di codice mostra come utilizzare `delete-relational-database`.

## AWS CLI

Per eliminare un database gestito

L'`delete-relational-databaseesempio` seguente elimina il database gestito specificato.

```
aws lightsail delete-relational-database \  
--relational-database-name Database-1
```

Output:

```
{  
  "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
    }
  ]
}

```

- Per API i dettagli, vedere [DeleteRelationalDatabase](#) in AWS CLI Command Reference.

## detach-static-ip

Il seguente esempio di codice mostra come utilizzare `detach-static-ip`.

### AWS CLI

Per scollegare un IP statico da un'istanza

L'esempio seguente rimuove l'IP statico `StaticIp-1` da qualsiasi istanza collegata.

```
aws lightsail detach-static-ip \
  --static-ip-name StaticIp-1
```

Output:

```

{
  "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
  }
]
}
```

- Per API i dettagli, vedere [DetachStaticIp](#) in AWS CLI Command Reference.

## get-active-names

Il seguente esempio di codice mostra come utilizzare `get-active-names`.

### AWS CLI

Per ottenere i nomi delle risorse attive

L'`get-active-names` esempio seguente restituisce i nomi delle risorse attive nella AWS regione configurata.

```
aws lightsail get-active-names
```

Output:

```
{
  "activeNames": [
    "WordPress-1",
```

```
    "StaticIp-1",
    "MEAN-1",
    "Plesk_Hosting_Stack_on_Ubuntu-1"
  ]
}
```

- Per API i dettagli, vedere [GetActiveNames](#) in AWS CLI Command Reference.

## get-auto-snapshots

Il seguente esempio di codice mostra come utilizzare `get-auto-snapshots`.

### AWS CLI

Per ottenere le istantanee automatiche disponibili per un'istanza

L'`get-auto-snapshots` esempio seguente restituisce, ad esempio, le istantanee automatiche disponibili. `WordPress-1`

```
aws lightsail get-auto-snapshots \
  --resource-name WordPress-1
```

Output:

```
{
  "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": []
  }
]
```

Per ulteriori informazioni, consulta [Conservazione di istantanee automatiche di istanze o dischi in Amazon Lightsail nella Lightsail Dev Guide](#).

- Per API i dettagli, consulta Command Reference. [GetAutoSnapshots](#) AWS CLI

## get-blueprints

Il seguente esempio di codice mostra come utilizzare `get-blueprints`.

### AWS CLI

Per ottenere i progetti per nuove istanze

L'`get-blueprints` esempio seguente mostra i dettagli su tutti i blueprint disponibili che possono essere utilizzati per creare nuove istanze in Amazon Lightsail.

```
aws lightsail get-blueprints
```

Output:

```
{
  "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-6g3gzfcih6dvu",
    "licenseUrl": "https://aws.amazon.com/marketplace/pp/
prodview-6g3gzfcih6dvu#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"
    },
    ...
}
]
}

```

- Per i API dettagli, consulta [GetBlueprints](#) Command Reference.AWS CLI

## get-bundles

Il seguente esempio di codice mostra come utilizzare `get-bundles`.

### AWS CLI

Per ottenere i pacchetti per le nuove istanze

L'esempio seguente mostra i dettagli su tutti i pacchetti disponibili che possono essere utilizzati per creare nuove istanze in Amazon Lightsail.

```
aws lightsail get-bundles
```

Output:

```
{
  "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"
  ]
}
```

```
    },  
    ...  
  }  
]  
}
```

- Per i API dettagli, consulta [GetBundles](#) Command Reference.AWS CLI

## get-cloud-formation-stack-records

Il seguente esempio di codice mostra come utilizzare `get-cloud-formation-stack-records`.

### AWS CLI

Per ottenere i record dello CloudFormation stack e gli stack associati

L'`get-cloud-formation-stack-records` esempio seguente mostra i dettagli sui record dello CloudFormation stack e sugli stack associati utilizzati per creare EC2 risorse Amazon da istantanee Amazon Lightsail esportate.

```
aws lightsail get-cloud-formation-stack-records
```

Output:

```
{  
  "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"
  }
}
]
}

```

- Per i API dettagli, consulta Command Reference. [GetCloudFormationStackRecords](#) AWS CLI

## get-disk-snapshot

Il seguente esempio di codice mostra come utilizzare `get-disk-snapshot`.

### AWS CLI

Per ottenere informazioni su un'istantanea del disco

L'`get-disk-snapshot` seguente visualizza i dettagli sullo snapshot `Disk-1-1566839161` del disco.

```
aws lightsail get-disk-snapshot \
  --disk-snapshot-name Disk-1-1566839161
```

Output:

```

{
  "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
  }
}
```

Per ulteriori informazioni, consultate il titolo della guida.

- Per API i dettagli, vedere [GetDiskSnapshot](#) in AWS CLI Command Reference.

## get-disk-snapshots

Il seguente esempio di codice mostra come utilizzare `get-disk-snapshots`.

### AWS CLI

Per ottenere informazioni su tutte le istantanee del disco

L'`get-disk-snapshots` seguente visualizza i dettagli su tutte le istantanee del disco nella regione AWS configurata.

```
aws lightsail get-disk-snapshots
```

Output:

```
{
  "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
  }
]
}

```

- Per API i dettagli, vedere [GetDiskSnapshots](#) in AWS CLI Command Reference.

## get-disk

Il seguente esempio di codice mostra come utilizzare `get-disk`.

## AWS CLI

Per ottenere informazioni su un disco di archiviazione a blocchi

L'get-disk esempio seguente visualizza i dettagli sul disco Disk-1.

```
aws lightsail get-disk \  
  --disk-name Disk-1
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consultate il titolo della guida.

- Per API i dettagli, vedere [GetDisk](#) in AWS CLI Command Reference.

## get-disks

Il seguente esempio di codice mostra come utilizzare get-disks.

## AWS CLI

Per ottenere informazioni su tutti i dischi di archiviazione a blocchi

L'get-disksempio seguente visualizza i dettagli su tutti i dischi nella regione AWS configurata.

```
aws lightsail get-disks
```

Output:

```
{
  "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"
    }
]
}

```

- Per API i dettagli, vedere [GetDisks](#) in AWS CLI Command Reference.

## get-domain

Il seguente esempio di codice mostra come utilizzare `get-domain`.

### AWS CLI

Per ottenere informazioni su un dominio

L'`get-domain` esempio seguente mostra i dettagli sul dominio `example.com`.

Nota: le API operazioni relative al dominio di Lightsail sono disponibili solo nella Regione. `us-east-1` AWS Se il CLI profilo è configurato per utilizzare una regione diversa, è necessario includere il parametro `--region us-east-1` altrimenti il comando avrà esito negativo.

```

aws lightsail get-domain \
  --domain-name example.com \
  --region us-east-1

```

Output:

```

{
  "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"
    }
]
}
}

```

- Per API i dettagli, vedere in Command Reference. [GetDomain](#) AWS CLI

## get-domains

Il seguente esempio di codice mostra come utilizzare `get-domains`.

### AWS CLI

Per ottenere informazioni su tutti i domini

L'`get-domains` esempio seguente mostra i dettagli su tutti i domini nella regione AWS configurata.

Nota: le API operazioni relative ai domini di Lightsail sono disponibili solo nella Regione. `us-east-1` AWS Se il CLI profilo è configurato per utilizzare una regione diversa, è necessario includere il `--region us-east-1` parametro o il comando avrà esito negativo.

```

aws lightsail get-domains \
  --region us-east-1

```

Output:

```

{
  "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"
    }
]
}
]
}

```

- Per API i dettagli, vedere [GetDomains](#) in AWS CLI Command Reference.

## get-export-snapshot-record

Il seguente esempio di codice mostra come utilizzare `get-export-snapshot-record`.

### AWS CLI

Per ottenere i record delle istantanee esportate su Amazon EC2

L'esempio seguente mostra i dettagli sulle istanze di Amazon Lightsail o sugli snapshot del disco esportati in Amazon. EC2

```
aws lightsail get-export-snapshot-records
```

Output:

```

{
  "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"
        }
    }
]
}

```

- Per i API dettagli, consulta [GetExportSnapshotRecord](#) Command Reference.AWS CLI

## get-instance-access-details

Il seguente esempio di codice mostra come utilizzare `get-instance-access-details`.

### AWS CLI

Per ottenere informazioni sulla chiave dell'host per un'istanza

L'`get-instance-access-detail`sempio seguente visualizza, ad esempio, le informazioni sulla chiave dell'host `WordPress_Multisite-1`.

```

aws lightsail get-instance-access-details \
  --instance-name WordPress_Multisite-1

```

## Output:

```
{
  "accessDetails": {
    "certKey": "ssh-rsa-cert-v01@openssh.com
AEXAMPLEEaC1yc2EtY2VydC12MDFAb3B1bnNzaC5jb20AAAAgNf076Dt3ppmPd0fPxZVMmS491aEAYYH9cHqAJ3fNML8
vEXAMPLE2eBWJyQvn7o1/
i0+s966h5sx8qUD791PB7q5UESd5VZGFtytrykfQJnjiwqe7EV5agzvjb1Lj26Fb37EKda9HVfC0u8pWbvky7Tyn9w29
+xMfQM9xVz0rXZmqx8uJidJpRgLCMTviofwQJU/
K1EXAMPLEAAAAAAAAABAAAALS00MzMzMDU4MzA40Dg1MTY2NjM40np6UWlndHk4UE1RSG9Stit0TG5QSEE9PQAAAAAsAAA
+LiB+ozNbUA0cdNL9Y67x7qPv/R7XhTc21+2A+8+GuVpK/Kz9dqDMKNAEXAMPLE+YYN
+tiXm7Y80gziK+7iDB7xUuQ4vghmn4+qgz9mKwYgWvVe2+0XLU7cnWPB7iU1HQg
+E3LUKrv4ZFw9pj7X2dFdNKfMxwWgI1ISWKimEXAMPLEEehjrf1Rqc/
QH6TpWcVpfcx8uvwVqdwTfKE/SfA5BCzbGGI1UmIUadh8nHcb5FamQ1hK7kECy47K/x9FMn/
KwmM7pCwJbSLDM07n9bnbvck6m8ZoB2N2YLMG5dW7BerEXAMPLEEobqfdtyYJHHe11EyyEJs1fWNU3D5JIGlgzcPAV
+Z1bQyUCZXf0os1Sa+HE85f0/
FRq9SVSBSHrmbeb0fr1PhgMzgSmqLeyhlbr6wwWIDbREXAMPLEJZ49H7RdQxdKyYrZPwvRgcr0qI2EL0tAajnpQQ8UZo
Aqter0xN5PhFL0J490WTacwCGRAjLhibAx7K1t/1ZXWo6c+ijq8c111327EXAMPLE/
e89GC89KcmKcxfGQniDAUgF8UqofIbq3Z0UgiAAYCVXc1I4L68NhVXyoWuQXPBRQSEXAMPLEWm74tDL9tFN3c7tSe/
Oz0cTR+4sAAAIPAAAAB3NzaC1yc2EAAAIAQnG/
L0DqiSnLrWhEox4aHqMgd0m0oLLAYx60QH9F0TM9EXAMPLE961rzSCMon7ZgsWnNl00wZQgDG
+rtJ4N0B7H0Vwns4ynUFbzNQ3qFGGeE31kxW1L41vV1iSy7sDk8aI0LmrKJi1LE1Qc1l8uboRlwoX0YEXAMPLEEaUCeX
+10+WEXAMPLEg6Y4U4ZvE2B3xyRdpvysb5TGFNTk5qPslacnVkoL0GsZZXMPLGJnG40BpQLLtpj9sNMxAgZPCAUjhkqk
+nx0904NUZ2pTWbVSUaV1gm6pug9xbwN01Im21t34JeLlKTqxcJ6zzS8W0c0KKpAm5c4hWkseMbyutS2jav/4hiS
+BhrYgptzfwe5qRXEXAMPLEEHZQr3YfGzYoBJ/
1LK3NHhx0ihhsfAYwMei0BFZT1F/7CT3IH4iitEkIgodI06/
Mw6UDqMPoZYQCK11EA6LFhYCOZG9drWcoRa741M4kY9TP028Za8gDMh1WpkXLq9Gixon50HP8aM/
sEXAMPLEEr2+fnkw+1Bto05L6+VKoPlXaGqZ/fBYEXAMPLEAMQHjnlM1JYNvtEEPhp+TNzXHzuixWf/
Ht04m0AVpXrzIDXaS102tXY=",
    "ipAddress": "192.0.2.0",
    "privateKey": "-----BEGIN RSA PRIVATE KEY-----
\nEXAMPLEBAAKCAQEa+AD3qeU2toBy505v7wnRLVo/tngVickL5+6Jf4tPrPeuoebM
\nfk1A+/ZTwe6uVBeneVWRhbcra8pH0CZ44sKnuxFeWoM7425S49uhW9+xCnWvR1Xw
\njrvKVm75Mu08p/cNvfWugrBuaPB65DspgxNn0fZWMVxpIpSq0SPWmSwQHV597d6C
\nrEXAMPLEEo8hJmqz2KFQ09X7fB21BruGgr9aXiNPmWmovYKqwFmrnFvR7odFmDecq
\n5EXAMPLE9dyU1ZsrWhGby77eYrVaF10GNGQ8qy1HGUiscquZ9NDIL49n4mXbfsTH
\n0EXAMPLE12ZqsflYnSaUYCwjE74qH8ECVpytQIDAQABaoIBAHeZV9Z58JHAjifz
\nCEXAMPLEEqC3do0VDgXS1kKI92qNo4z2VcUEho878paCuVvXVHcCGgSnGeyIh2tN
\nMEXAMPLESohR427BhH3YLA+3Z5SivnejbTgYPfLC37B8khTaYqkqMvdZiFVZK5qn
\nIEXAMPLEM93oF9eSZCjcLKB/jGhsfb0eCDMP8BshHE2beuqzVMok1Dx0nvoP3+Fp
\nAEXAMPLESq6pDpCo9YVUX8g1u3Ro9cP12LXHDy+oVEY5KhbZQJ7VU1I72W0vppWW
\n0EXAMPLEkgYlq7p6qYtYcSgTEjz14gDiMfQ7SyHB3alkIoN0NQ9ZPaWHyJvymeud
\noQTNuz0CgYEA/LFWNTEZrzdzdR1kJmyNrmAermU0B6utyNENChAlHGSHkB+11VSh
\nbEXAMPLEQo9ooUeW5Ux03YwacZLoDT1mwxw1Ptc1+PNycZoLe1fE9UdARrdmGTob
```

```

\n817CPLSXp3xuR8VqSp2fnIc7hfiQs/NrPX9gm/E0rB0we0RKyDSzWScCgYEA+z/r
\niob+nJZq0Ybn0SuP6oMULP4vnWniWj8MIhUJU53LwSAM8DeJd0NKDdkui0d52aAL
\nVgn7nLo88rVWKhJwVc4tu/rNgZLcR3bP4+kL6zand0KQnMLy0zNA2Ys26aa5udH1\nqWl0WTt9WEm/
h10ndC1kn0MectrvsG17b38y5sMCgYEA54NiRGgz8oCPW6GN/FZA
\nKEXAMPLE5tw34GEH3UxlC9n3CejDaQmcz0ATwX4nIwRZDEqWyYZcS0btg1jhGiBD\nYEXAMPLEkc8Z71L/
agZEAaVCEog9FqfSqwB
+XTfoKh8qur74X1yCu9p6gof1q6k9\nEXAMPLEchJcNN0g4ETIfMkCgYBdVORRHE4mqvWp0dzA7v66FdEz2YSkjAXKk
\naEXAMPLE8Z/8yBSmuBv1Qv03XA12my462uB92uzzGAuW
+1yBc2Kn1sXqYTy0y1z0\nngEXAMPLEBogjw4MqHKL1bPKMHYQU8/
q24PaYgzHPzy13wLH6pTYf1XqLHdE2D6Vv\nnyEXAMPLEgQC3i/
kVVhky/2XRwRV1C7J02Bg3QGTx38hpmDa5IuofKANjA+Wa3/zy\nnbEXAMPLE6ytQgD9GN/YtBq+uh0
+2ZkvXPL+CWRi0ZRXPwYDBBFU9Cw0AuWWG1L8\nnwEXAMPLExMlcysRgcWB9RNgf3Au0pFd2i6XT/
riNsvvKpmJ+VooU8g==\n-----END RSA PRIVATE KEY-----\n",
    "protocol": "ssh",
    "instanceName": "WordPress_Multisite-1",
    "username": "bitnami",
    "hostKeys": [
        {
            "algorithm": "ssh-rsa",
            "publicKey":
"AEXAMPLEaC1yc2EAAAADAQABAAQCoer9ieZTjQ3pXCHczuAYZFj1F7t
+uBkXuqeGMRex78pCvmS+DiEXAMPLEuEj1Q8dcKhrQL4HpXbD9dosVCTaJnJwb4MQqsuSVFdHFzy3guP
+BKc1WqtxJEXAMPLEsBGqZz1rIv6a9bTA0TCplZ8AD+hSRTaSXXqg6FT
+Qf16IktH0X1Ms7xIEXAMPLEmNtjCpzZiGXDHzytoMvUgwa8uHPp440g36EUu4VqQxoUHPJKoXvcQizyk3K8ym0hP0Tp
0t6y9HwvykEXAMPLEAfbKjBR42+u6+0S1kr4d339q2U1sTDytJhhs8HUe11wTfGRfp",
            "witnessedAt": 1570744377.699,
            "fingerprintSHA1": "SHA1:GEXAMPLEMoYgUg0ucadqU9Bt3Lk",
            "fingerprintSHA256": "SHA256:IEEXAMPLEcB5vgxnAUoJawbdZ
+MwELhIp6FUxuwq/LIU"
        },
        {
            "algorithm": "ssh-ed25519",
            "publicKey":
"AEXAMPLEaC1lZDI1NTE5AAAAIC1gwGPDfGa0NxEXAMPLEJX3UNap781QxHQmn8nzlrUv",
            "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"
    }
  ]
}
}

```

- Per API i dettagli, vedere [GetInstanceAccessDetails](#) in AWS CLI Command Reference.

## get-instance-metric-data

Il seguente esempio di codice mostra come utilizzare `get-instance-metric-data`.

### AWS CLI

Per ottenere dati metrici per un'istanza

L'`get-instance-metric-data` esempio seguente restituisce la percentuale media di CPUUtilization ogni 7200 secondo (2 ore) tra 1571342400 e 1571428800 per esempio MEAN-1.

Si consiglia di utilizzare un convertitore di tempo unix per identificare l'ora di inizio e di fine.

```

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

```

Output:

```

{
  "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"
    }
]
}

```

- Per API i dettagli, vedere [GetInstanceMetricData](#) in AWS CLI Command Reference.

## get-instance-port-states

Il seguente esempio di codice mostra come utilizzare `get-instance-port-states`.

### AWS CLI

Per ottenere informazioni sul firewall per un'istanza

L'`get-instance-port-states` esempio seguente restituisce le porte firewall configurate ad esempio `MEAN-1`.

```

aws lightsail get-instance-port-states \
  --instance-name MEAN-1

```

Output:

```

{
  "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"
}
]
```

- Per API i dettagli, vedere [GetInstancePortStates](#) in AWS CLI Command Reference.

## get-instance-snapshot

Il seguente esempio di codice mostra come utilizzare `get-instance-snapshot`.

### AWS CLI

Per ottenere informazioni su un'istantanea dell'istanza specificata

L'`get-instance-snapshot` esempio seguente visualizza i dettagli sullo snapshot dell'istanza specificata.

```
aws lightsail get-instance-snapshot \
  --instance-snapshot-name MEAN-1-1571419854
```

Output:

```
{
  "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
}
}

```

- Per API i dettagli, vedere [GetInstanceSnapshot](#) in AWS CLI Command Reference.

## get-instance-snapshots

Il seguente esempio di codice mostra come utilizzare `get-instance-snapshots`.

### AWS CLI

Per ottenere informazioni su tutte le istantanee delle istanze

L'`get-instance-snapshots` esempio seguente mostra i dettagli su tutte le istantanee delle istanze nella regione AWS configurata.

```
aws lightsail get-instance-snapshots
```

Output:

```

{
  "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
}
]
}
```

- Per API i dettagli, vedere [GetInstanceSnapshots](#) in AWS CLI Command Reference.

## get-instance-state

Il seguente esempio di codice mostra come utilizzare `get-instance-state`.

### AWS CLI

Per ottenere informazioni sullo stato di un'istanza

L'`get-instance-state` esempio seguente restituisce lo stato dell'istanza specificata.

```
aws lightsail get-instance-state \  
  --instance-name MEAN-1
```

Output:

```
{  
  "state": {  
    "code": 16,  
    "name": "running"  
  }  
}
```

- Per API i dettagli, vedere [GetInstanceState](#) in AWS CLI Command Reference.

## get-instance

Il seguente esempio di codice mostra come utilizzare `get-instance`.

### AWS CLI

Per ottenere informazioni su un'istanza

L'`get-instance` esempio seguente mostra i dettagli sull'istanza `MEAN-1`.

```
aws lightsail get-instance \  
  --instance-name MEAN-1
```

Output:

```
{
```

```
"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"
}
}

```

- Per API i dettagli, vedere [GetInstance](#) in AWS CLI Command Reference.

## get-instances

Il seguente esempio di codice mostra come utilizzare `get-instances`.

### AWS CLI

Per ottenere informazioni su tutte le istanze

L'get-instancesesempio seguente mostra i dettagli su tutte le istanze nella regione AWS configurata.

```
aws lightsail get-instances
```

Output:

```
{
  "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"
}
]
}
```

- Per API i dettagli, vedere [GetInstances](#) in AWS CLI Command Reference.

## get-key-pair

Il seguente esempio di codice mostra come utilizzare `get-key-pair`.

### AWS CLI

Per ottenere informazioni su una key pair

L'`get-key-pair` esempio seguente visualizza i dettagli sulla coppia di chiavi specificata.

```
aws lightsail get-key-pair \  
  --key-pair-name MyKey1
```

Output:

```
{  
  "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"  
  }  
}
```

- Per API i dettagli, vedere [GetKeyPair](#) in AWS CLI Command Reference.

## get-key-pairs

Il seguente esempio di codice mostra come utilizzare `get-key-pairs`.

### AWS CLI

Per ottenere informazioni su tutte le coppie di chiavi

L'`get-key-pairs` esempio seguente mostra i dettagli su tutte le coppie di chiavi nella AWS regione configurata.

```
aws lightsail get-key-pairs
```

Output:

```
{
```

```

    "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"
      }
    ]
  }

```

- Per API i dettagli, vedere [GetKeyPairs](#) in AWS CLI Command Reference.

## get-load-balancer-tls-certificates

Il seguente esempio di codice mostra come utilizzare `get-load-balancer-tls-certificates`.

### AWS CLI

Per ottenere informazioni sui TLS certificati per un sistema di bilanciamento del carico

L'`get-load-balancer-tls-certificates` esempio seguente visualizza i dettagli sui TLS certificati per il sistema di bilanciamento del carico specificato.

```

aws lightsail get-load-balancer-tls-certificates \
  --load-balancer-name LoadBalancer-1

```

Output:

```

{
  "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"
    ]
}
]
}

```

- Per API i dettagli, vedere [GetLoadBalancerTlsCertificates](#) in AWS CLI Command Reference.

## get-load-balancer

Il seguente esempio di codice mostra come utilizzare `get-load-balancer`.

### AWS CLI

Per ottenere informazioni su un sistema di bilanciamento del carico

L'esempio seguente mostra i dettagli sul sistema di bilanciamento del carico specificato.

```
aws lightsail get-load-balancer \
  --load-balancer-name LoadBalancer-1
```

Output:

```
{
  "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"
    }
  }
}
```

- Per API i dettagli, vedere [GetLoadBalancer](#) in AWS CLI Command Reference.

## get-load-balancers

Il seguente esempio di codice mostra come utilizzare `get-load-balancers`.

### AWS CLI

Per ottenere informazioni su tutti i sistemi di bilanciamento del carico

L'esempio seguente mostra i dettagli su tutti i sistemi di bilanciamento del carico nella regione configurata. AWS

```
aws lightsail get-load-balancers
```

Output:

```
{
```

```
"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"
        }
    ]
}
```

- Per API i dettagli, vedere [GetLoadBalancers](#) in AWS CLI Command Reference.

## get-operation

Il seguente esempio di codice mostra come utilizzare `get-operation`.

### AWS CLI

Per ottenere informazioni su una singola operazione

L'`get-operation` esempio seguente visualizza i dettagli sull'operazione specificata.

```
aws lightsail get-operation \
  --operation-id e5700e8a-daf2-4b49-bc01-3EXAMPLE910a
```

Output:

```
{
  "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
  }
}
```



- Per API i dettagli, vedere [GetOperation](#) in AWS CLI Command Reference.

## get-operations-for-resource

Il seguente esempio di codice mostra come utilizzare `get-operations-for-resource`.

### AWS CLI

Per ottenere tutte le operazioni relative a una risorsa

L'`get-operations-for-resource` esempio seguente mostra i dettagli su tutte le operazioni per la risorsa specificata.

```
aws lightsail get-operations-for-resource \  
  --resource-name LoadBalancer-1
```

Output:

```
{  
  "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-1EXAMPLEdac2",
    "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
},
...
}
]
}

```

- Per API i dettagli, vedere [GetOperationsForResource](#) in AWS CLI Command Reference.

## get-operations

Il seguente esempio di codice mostra come utilizzare `get-operations`.

### AWS CLI

Per ottenere informazioni su tutte le operazioni

L'output di `get-operations` seguente mostra i dettagli su tutte le operazioni nella AWS regione configurata.

```
aws lightsail get-operations
```

## Output:

```
{
  "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
    },
    ...
}
]
```

- Per API i dettagli, vedere [GetOperations](#) in AWS CLI Command Reference.

## get-regions

Il seguente esempio di codice mostra come utilizzare `get-regions`.

### AWS CLI

Per ottenere tutte le AWS regioni per Amazon Lightsail

L'`get-regions` esempio seguente mostra i dettagli su tutte le AWS regioni per Amazon Lightsail.

```
aws lightsail get-regions
```

Output:

```
{
  "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": []
    },
    ...
}
]
}

```

- Per API i dettagli, consulta AWS CLI Command [GetRegionsReference](#).

## get-relational-database-blueprints

Il seguente esempio di codice mostra come utilizzare `get-relational-database-blueprints`.

### AWS CLI

Per ottenere i progetti per nuovi database relazionali

L'esempio seguente mostra i dettagli su tutti i blueprint di database relazionali disponibili che possono essere utilizzati per creare nuovi database relazionali in Amazon Lightsail.

```
aws lightsail get-relational-database-blueprints
```

Output:

```

{
  "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
  }
]
```

```
}
```

- Per i API dettagli, consulta Command Reference. [GetRelationalDatabaseBlueprints](#) AWS CLI

## get-relational-database-bundles

Il seguente esempio di codice mostra come utilizzare `get-relational-database-bundles`.

### AWS CLI

Per ottenere i pacchetti per i nuovi database relazionali

L'`get-relational-database-bundles` esempio seguente mostra i dettagli su tutti i pacchetti di database relazionali disponibili che possono essere utilizzati per creare nuovi database relazionali in Amazon Lightsail. Tieni presente che la risposta non include pacchetti inattivi perché il `--include-inactive` flag non è specificato nel comando. Non è possibile utilizzare pacchetti inattivi per creare nuovi database relazionali.

```
aws lightsail get-relational-database-bundles
```

Output:

```
{
  "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
  }
]
}
```

Per ulteriori informazioni, consulta [Creazione di un database in Amazon Lightsail nella Amazon Lightsail Developer Guide](#).

- Per i API dettagli, consulta [GetRelationalDatabaseBundles](#) Command Reference.AWS CLI

## **get-relational-database-events**

Il seguente esempio di codice mostra come utilizzare `get-relational-database-events`.

AWS CLI

Per ottenere gli eventi per un database relazionale

L'get-relational-database-eventsempio seguente visualizza i dettagli sugli eventi delle ultime 17 ore (1020 minuti) per il database relazionale specificato.

```
aws lightsail get-relational-database-events \  
  --relational-database-name Database-1 \  
  --duration-in-minutes 1020
```

Output:

```
{  
  "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"  
      ]  
    }  
  ]  
}
```

- Per API i dettagli, vedere [GetRelationalDatabaseEvents](#) in AWS CLI Command Reference.

## get-relational-database-log-events

Il seguente esempio di codice mostra come utilizzare get-relational-database-log-events.

AWS CLI

Per ottenere gli eventi di registro per un database relazionale

L'get-relational-database-log-events esempio seguente visualizza i dettagli sul registro specificato tra 1570733176 e 1571597176 per il database relazionale. Database1 Le informazioni restituite sono configurate per iniziare dahead.

Si consiglia di utilizzare un convertitore di tempo unix per identificare l'ora di inizio e di fine.

```
aws lightsail get-relational-database-log-events \  
  --relational-database-name Database1 \  
  --log-stream-name error \  
  --start-from-head \  
  --start-time 1570733176 \  
  --end-time 1571597176
```

Output:

```
{  
  "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":
"eEXAMPLEZXJUZXh0IjoiZnRwb3F3cUpRS1Q5NndMYThxe1RUZ1FhR3J6c2dKWEEvM2kvajZMzZVVVWpqRDN0YjFXTj
"nextForwardToken":
"eEXAMPLEZXJUZXh0IjoiT09Lb0Z6ZFRJbHhaNEQ5N2tPbkkwRmwwNUxPZjFTbFFwUk1Qbz1SaWgvMwVXbEk4aG56VH
}

```

- Per API i dettagli, vedere [GetRelationalDatabaseLogEvents](#) in AWS CLI Command Reference.

## get-relational-database-log-streams

Il seguente esempio di codice mostra come utilizzare `get-relational-database-log-streams`.

### AWS CLI

Per ottenere i flussi di log per un database relazionale

L'`get-relational-database-log-streams` esempio seguente restituisce tutti i flussi di log disponibili per il database relazionale specificato.

```
aws lightsail get-relational-database-log-streams \
--relational-database-name Database1
```

Output:

```
{
```

```
"logStreams": [  
  "audit",  
  "error",  
  "general",  
  "slowquery"  
]  
}
```

- Per API i dettagli, vedere [GetRelationalDatabaseLogStreams](#) in AWS CLI Command Reference.

## get-relational-database-master-user-password

Il seguente esempio di codice mostra come utilizzare `get-relational-database-master-user-password`.

### AWS CLI

Per ottenere la password dell'utente principale per un database relazionale

L'`get-relational-database-master-user-password` esempio seguente restituisce informazioni sulla password dell'utente principale per il database relazionale specificato.

```
aws lightsail get-relational-database-master-user-password \  
  --relational-database-name Database-1
```

Output:

```
{  
  "masterUserPassword": "VEXAMPLEec.9qvX,_t<)Wkf)kwboM,>2",  
  "createdAt": 1571259453.959  
}
```

- Per API i dettagli, vedere [GetRelationalDatabaseMasterUserPassword](#) in AWS CLI Command Reference.

## get-relational-database-metric-data

Il seguente esempio di codice mostra come utilizzare `get-relational-database-metric-data`.

## AWS CLI

Per ottenere dati metrici per un database relazionale

L'get-relational-database-metric-data esempio seguente restituisce la somma del conteggio della metrica DatabaseConnections nel periodo di 24 ore (86400secondi) tra 1570733176 e 1571597176 per il database relazionale. Database1

Si consiglia di utilizzare un convertitore di tempo unix per identificare l'ora di inizio e di fine.

```
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
```

Output:

```
{  
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [GetRelationalDatabaseMetricData](#) in AWS CLI Command Reference.

## get-relational-database-parameters

Il seguente esempio di codice mostra come utilizzare `get-relational-database-parameters`.

### AWS CLI

Per ottenere i parametri per un database relazionale

L'get-relational-database-parameteresempio seguente restituisce informazioni su tutti i parametri disponibili per il database relazionale specificato.

```
aws lightsail get-relational-database-parameters \  
--relational-database-name Database-1
```

Output:

```
{  
  "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"
    },
    ...
}
]
}

```

Per ulteriori informazioni, consulta [Aggiornamento dei parametri del database in Amazon Lightsail](#) nella Lightsail Dev Guide.

- Per i API dettagli, consulta [GetRelationalDatabaseParameters](#) Command Reference.AWS CLI

## get-relational-database-snapshot

Il seguente esempio di codice mostra come utilizzare `get-relational-database-snapshot`.

### AWS CLI

Per ottenere informazioni su uno snapshot di un database relazionale

L'`get-relational-database-snapshot` seguente visualizza i dettagli sullo snapshot del database relazionale specificato.

```
aws lightsail get-relational-database-snapshot \
  --relational-database-snapshot-name Database-1-1571350042
```

Output:

```
{
  "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"
  }
}

```

- Per API i dettagli, vedere [GetRelationalDatabaseSnapshot](#) in AWS CLI Command Reference.

## get-relational-database-snapshots

Il seguente esempio di codice mostra come utilizzare `get-relational-database-snapshots`.

### AWS CLI

Per ottenere informazioni su tutte le istantanee dei database relazionali

L'esempio seguente visualizza i dettagli su tutte le istantanee del database relazionale nella regione configurata. AWS

```
aws lightsail get-relational-database-snapshots
```

Output:

```

{
  "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"
}
]
```

```
}
```

- Per API i dettagli, vedere [GetRelationalDatabaseSnapshots](#) in AWS CLI Command Reference.

## get-relational-database

Il seguente esempio di codice mostra come utilizzare `get-relational-database`.

### AWS CLI

Per ottenere informazioni su un database relazionale

L'`get-relational-database` esempio seguente visualizza i dettagli sul database relazionale specificato.

```
aws lightsail get-relational-database \  
  --relational-database-name Database-1
```

Output:

```
{  
  "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": "ls-9EXAMPLE8ad863723b62ccEXAMPLEa6e794ae0d2.czowadgeezqi.us-
west-2.rds.amazonaws.com"
    },
    "pendingMaintenanceActions": []
  }
}

```

- Per API i dettagli, vedere [GetRelationalDatabase](#) in AWS CLI Command Reference.

## get-relational-databases

Il seguente esempio di codice mostra come utilizzare `get-relational-databases`.

### AWS CLI

Per ottenere informazioni su tutti i database relazionali

L'esempio seguente mostra i dettagli su tutti i database relazionali nella regione AWS configurata.

```
aws lightsail get-relational-databases
```

Output:

```

{
  "relationalDatabases": [
    {
      "name": "MySQL",
      "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/8529020c-3ab9-4d51-92af-5EXAMPLE8979",

```

```

    "supportCode": "6EXAMPLE3362/
ls-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.czowadgezqi.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":
"ls-3EXAMPLEb4ffffb5cec056220c734713eEXAMPLEd.czowadgeezqi.us-
west-2.rds.amazonaws.com"
    },
    "pendingMaintenanceActions": []
}
]
}

```

- Per API i dettagli, vedere [GetRelationalDatabases](#) in AWS CLI Command Reference.

## get-static-ip

Il seguente esempio di codice mostra come utilizzare `get-static-ip`.

## AWS CLI

Per ottenere informazioni su un IP statico

L'`get-static-ip` seguente visualizza i dettagli sull'IP statico specificato.

```
aws lightsail get-static-ip \  
  --static-ip-name StaticIp-1
```

Output:

```
{  
  "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  
  }  
}
```

- Per API i dettagli, vedere [GetStaticIp](#) in AWS CLI Command Reference.

## get-static-ips

Il seguente esempio di codice mostra come utilizzare `get-static-ips`.

### AWS CLI

Per ottenere informazioni su tutto ciò che è statico IPs

L'`get-static-ips` seguente mostra i dettagli su tutte le informazioni statiche IPs nella AWS regione configurata.

```
aws lightsail get-static-ips
```



**Output:**

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [GetStaticIps](#) in AWS CLI Command Reference.

**is-vpc-peered**

Il seguente esempio di codice mostra come utilizzare `is-vpc-peered`.

## AWS CLI

Per determinare se il tuo cloud privato virtuale Amazon Lightsail è peer-to-peer

L'`is-vpc-peered` seguente restituisce lo stato di peering del VPC cloud privato virtuale Amazon Lightsail () per la regione specificata. AWS

```
aws lightsail is-vpc-peered \  
  --region us-west-2
```

Output:

```
{  
  "isPeered": true  
}
```

- Per API i dettagli, consulta Command [IsVpcPeered](#) Reference AWS CLI .

## open-instance-public-ports

Il seguente esempio di codice mostra come utilizzare `open-instance-public-ports`.

### AWS CLI

Per aprire le porte del firewall per un'istanza

L'`open-instance-public-ports` seguente apre la TCP porta 22 sull'istanza specificata.

```
aws lightsail open-instance-public-ports \  
  --instance-name MEAN-2 \  
  --port-info fromPort=22,protocol=TCP,toPort=22
```

Output:

```
{  
  "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
  }
}

```

- Per API i dettagli, vedere [OpenInstancePublicPorts](#) in AWS CLI Command Reference.

## peer-vpc

Il seguente esempio di codice mostra come utilizzare `peer-vpc`.

### AWS CLI

Per eseguire il peering del cloud privato virtuale Amazon Lightsail

L'`peer-vpc` esempio seguente esegue il peer del VPC cloud privato virtuale Amazon Lightsail () per la regione specificata. AWS

```

aws lightsail peer-vpc \
  --region us-west-2

```

Output:

```

{
  "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
  }
}
```

- Per API i dettagli, consulta Command [PeerVpcReference](#) AWS CLI .

## reboot-instance

Il seguente esempio di codice mostra come utilizzare `reboot-instance`.

### AWS CLI

Per riavviare un'istanza

L'`reboot-instance` esempio seguente riavvia l'istanza specificata.

```
aws lightsail reboot-instance \
  --instance-name MEAN-1
```

Output:

```
{
  "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
    }
  ]
}
```

```
]
}
```

- Per API i dettagli, vedere [RebootInstance](#) in AWS CLI Command Reference.

## reboot-relational-database

Il seguente esempio di codice mostra come utilizzare `reboot-relational-database`.

### AWS CLI

Per riavviare un database relazionale

L'`reboot-relational-database` esempio seguente riavvia il database relazionale specificato.

```
aws lightsail reboot-relational-database \
  --relational-database-name Database-1
```

Output:

```
{
  "operations": [
    {
      "id": "e4c980c0-3137-496c-9c91-1EXAMPLEdec2",
      "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
    }
  ]
}
```

- Per API i dettagli, vedere [RebootRelationalDatabase](#) in AWS CLI Command Reference.

## release-static-ip

Il seguente esempio di codice mostra come utilizzare `release-static-ip`.

### AWS CLI

Per eliminare un IP statico

L'`release-static-ip` seguente elimina l'IP statico specificato.

```
aws lightsail release-static-ip \  
  --static-ip-name StaticIp-1
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

- Per API i dettagli, vedere [ReleaseStaticIp](#) in AWS CLI Command Reference.

## start-instance

Il seguente esempio di codice mostra come utilizzare `start-instance`.

### AWS CLI

Per avviare un'istanza

L'`start-instance`esempio seguente avvia l'istanza specificata.

```
aws lightsail start-instance \  
  --instance-name WordPress-1
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

- Per API i dettagli, vedere [StartInstance](#)in AWS CLI Command Reference.

## **start-relational-database**

Il seguente esempio di codice mostra come utilizzare `start-relational-database`.

AWS CLI

Per avviare un database relazionale

L'`start-relational-database`esempio seguente avvia il database relazionale specificato.

```
aws lightsail start-relational-database \  
  --relational-database-name Database-1
```

**Output:**

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [StartRelationalDatabase](#) in AWS CLI Command Reference.

**stop-instance**

Il seguente esempio di codice mostra come utilizzare `stop-instance`.

**AWS CLI**

Per interrompere un'istanza

L'`stop-instance` esempio seguente arresta l'istanza specificata.

```
aws lightsail stop-instance \
--instance-name WordPress-1
```

**Output:**

```
{
  "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
  }
]
}

```

- Per API i dettagli, vedere [StopInstance](#) in AWS CLI Command Reference.

## stop-relational-database

Il seguente esempio di codice mostra come utilizzare `stop-relational-database`.

### AWS CLI

Per interrompere un database relazionale

L'`stop-relational-database` esempio seguente arresta il database relazionale specificato.

```

aws lightsail stop-relational-database \
  --relational-database-name Database-1

```

Output:

```

{
  "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
    }
  ]
}

```

- Per API i dettagli, vedere [StopRelationalDatabase](#) in AWS CLI Command Reference.

## unpeer-vpc

Il seguente esempio di codice mostra come utilizzare `unpeer-vpc`.

### AWS CLI

Per rendere anonimo il cloud privato virtuale Amazon Lightsail

L'`unpeer-vpc` esempio seguente annulla il peer del cloud privato virtuale Amazon Lightsail VPC () per la regione specificata. AWS

```

aws lightsail unpeer-vpc \
  --region us-west-2

```

Output:

```

{
  "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  
  }  
}
```

- Per i API dettagli, consulta [UnpeerVpcCommand Reference.AWS CLI](#)

## Esempi di Macie che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Macie.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **describe-buckets**

Il seguente esempio di codice mostra come utilizzare `describe-buckets`.

AWS CLI

Per interrogare i dati su uno o più bucket S3 monitorati e analizzati da Amazon Macie per il tuo account

L'esempio seguente interroga i metadati per tutti i bucket S3 i cui nomi iniziano con MY-S3 e si trovano nella regione corrente. AWS

```
aws macie2 describe-buckets \  
  --criteria '{"bucketName":{"prefix":"my-S3"}}'
```

## Output:

```
{
  "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  
  }  
]  
}
```

Per ulteriori informazioni, consulta [Filtrare l'inventario dei bucket S3](#) nella Guida per l'utente di Amazon Macie.

- Per i API dettagli, consulta [DescribeBuckets](#) Command Reference.AWS CLI

## Esempi di Amazon Managed Grafana utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Managed Grafana. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **list-workspaces**

Il seguente esempio di codice mostra come utilizzare `list-workspaces`.

AWS CLI

Per elencare le aree di lavoro per l'account nella regione specificata dalla credenziale dell'utente

L'`list-workspaces` esempio seguente elenca le aree di lavoro Grafana per la regione dell'account.



**aws grafana list-workspaces**

Output:

```
{
  "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"
    }
  ]
}
```

```
}
```

- Per API i dettagli, vedere [ListWorkspaces](#) in AWS CLI Command Reference.

## MediaConnect esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with MediaConnect.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

### Azioni

#### **add-flow-outputs**

Il seguente esempio di codice mostra come utilizzare `add-flow-outputs`.

AWS CLI

Per aggiungere output a un flusso

L'`add-flow-outputs` esempio seguente aggiunge gli output al flusso specificato.

```
aws mediaconnect add-flow-outputs \  
--flow-arn arn:aws:mediaconnect: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
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Aggiungere output a un flusso nella Guida per l'utente di AWS MediaConnect Elemental](#).

- Per API i dettagli, consulta AWS CLI Command [AddFlowOutputs](#) Reference.

## create-flow

Il seguente esempio di codice mostra come utilizzare `create-flow`.

### AWS CLI

Per creare un flusso

L'create-flowesempio seguente crea un flusso con la configurazione specificata.

```
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
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Creazione di un flusso](#) nella Guida per l' MediaConnect utente di AWS Elemental.

- Per API i dettagli, consulta [CreateFlow AWS CLI Command Reference](#).

## delete-flow

Il seguente esempio di codice mostra come utilizzare `delete-flow`.

### AWS CLI

Per eliminare un flusso

L'`delete-flow` esempio seguente elimina il flusso specificato.

```
aws mediaconnect delete-flow \  
  --flow-arn arn:aws:mediaconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

Output:

```
{  
  "FlowArn": "arn:aws:mediaconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",  
  "Status": "DELETING"  
}
```

Per ulteriori informazioni, consulta [Eliminazione di un flusso](#) nella Guida per l'utente di AWS MediaConnect Elemental.

- Per API i dettagli, consulta AWS CLI Command [DeleteFlow](#) Reference.

## describe-flow

Il seguente esempio di codice mostra come utilizzare `describe-flow`.

### AWS CLI

Per visualizzare i dettagli di un flusso

L'`describe-flow` esempio seguente visualizza i dettagli del flusso specificato, ad esempio la zona di disponibilità ARN, lo stato, l'origine, i permessi e gli output.

```
aws mediaconnect describe-flow \  
  --flow-arn arn:aws:mediaconnect:us-  
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

## Output:

```
{
  "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:mediaconnect: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"
  }
}
}

```

Per ulteriori informazioni, consulta [Visualizzazione dei dettagli di un flusso nella Guida per l'utente di AWS MediaConnect Elemental](#).

- Per API i dettagli, consulta [DescribeFlow AWS CLI Command Reference](#).

## grant-flow-entitlements

Il seguente esempio di codice mostra come utilizzare `grant-flow-entitlements`.

### AWS CLI

Per concedere un'autorizzazione su un flusso

L'`grant-flow-entitlements` seguente concede il diritto al flusso esistente specificato di condividere i tuoi contenuti con un altro account. AWS

```

aws mediaconnect grant-flow-entitlements \
  --flow-arn arn:aws:mediaconnect: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

```

**Output:**

```
{
  "Entitlements": [
    {
      "Name": "AnyCompany_Entitlement",
      "EntitlementArn": "arn:aws:mediacconnect: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:mediacconnect: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"
}
```

Per ulteriori informazioni, consulta [Concessione di un diritto su un flusso nella Guida per l'utente di Elemental.AWS MediaConnect](#)

- Per i API dettagli, consulta [GrantFlowEntitlements](#) Command Reference.AWS CLI

**list-entitlements**

Il seguente esempio di codice mostra come utilizzare `list-entitlements`.



## AWS CLI

Per visualizzare un elenco di diritti

L'`list-entitlements`esempio seguente visualizza un elenco di tutti i diritti concessi all'account.

```
aws mediaconnect list-entitlements
```

Output:

```
{
  "Entitlements": [
    {
      "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:MyEntitlement",
      "EntitlementName": "MyEntitlement"
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione [ListEntitlements](#)Riferimento AWS elementare MediaConnect API.

- Per API i dettagli, vedere [ListEntitlements](#)in AWS CLI Command Reference.

## list-flows

Il seguente esempio di codice mostra come utilizzare `list-flows`.

### AWS CLI

Per visualizzare un elenco di flussi

L'`list-flows`esempio seguente visualizza un elenco di flussi.

```
aws mediaconnect list-flows
```

Output:

```
{
  "Flows": [
```

```

    {
      "Status": "STANDBY",
      "SourceType": "OWNED",
      "AvailabilityZone": "us-west-2a",
      "Description": "NYC awards show",
      "Name": "AwardsShow",
      "FlowArn": "arn:aws:mediacconnect: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:mediacconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Visualizzazione di un elenco di flussi](#) nella Guida per l'MediaConnect utente di AWS Elemental.

- Per API i dettagli, consulta [ListFlows AWS CLI](#) Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una MediaConnect risorsa

L'`list-tags-for-resource` esempio seguente visualizza le chiavi e i valori dei tag associati alla MediaConnect risorsa specificata.

```

aws mediacconnect list-tags-for-resource \
  --resource-arn arn:aws:mediacconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame

```

Output:

```
{
  "Tags": {
    "region": "west",
    "stage": "prod"
  }
}
```

Per ulteriori informazioni, consulta [ListTagsForResource](#), [TagResource](#), [UntagResource](#) nel MediaConnect API Riferimento AWS elementare.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## remove-flow-output

Il seguente esempio di codice mostra come utilizzare `remove-flow-output`.

### AWS CLI

Per rimuovere un output da un flusso

L'`remove-flow-output` esempio seguente rimuove un output dal flusso specificato.

```
aws mediaconnect remove-flow-output \
  --flow-arn arn:aws:mediaconnect:us-east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
  --output-arn arn:aws:mediaconnect:us-east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC
```

Output:

```
{
  "FlowArn": "arn:aws:mediaconnect:us-east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
  "OutputArn": "arn:aws:mediaconnect:us-east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC"
}
```

Per ulteriori informazioni, vedete [Rimuovere gli output da un flusso nella Guida MediaConnect](#) utente di AWS Elemental.

- Per API i dettagli, consulta AWS CLI Command [RemoveFlowOutput](#) Reference.

## revoke-flow-entitlement

Il seguente esempio di codice mostra come utilizzare `revoke-flow-entitlement`.

### AWS CLI

Per revocare un diritto

L'esempio seguente revoca un'autorizzazione sul flusso specificato.

```
aws mediaconnect revoke-flow-entitlement \
  --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
  --entitlement-arn arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement
```

Output:

```
{
  "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
  "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement"
}
```

Per ulteriori informazioni, consulta [Revoca di un diritto](#) nella Guida per l'utente di Elemental.AWS MediaConnect

- Per i API dettagli, consulta Command Reference. [RevokeFlowEntitlement](#) AWS CLI

## start-flow

Il seguente esempio di codice mostra come utilizzare `start-flow`.

### AWS CLI

Per avviare un flusso

L'esempio seguente avvia il flusso specificato.

```
aws mediaconnect start-flow \
```

```
--flow-arn arn:aws:mediacconnect:us-east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

Questo comando non produce alcun output. Output:

```
{  
  "FlowArn": "arn:aws:mediacconnect:us-east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",  
  "Status": "STARTING"  
}
```

Per ulteriori informazioni, consulta [Avvio di un flusso](#) nella Guida per l' MediaConnect utente di AWS Elemental.

- Per API i dettagli, consulta [StartFlow AWS CLI](#) Command Reference.

## stop-flow

Il seguente esempio di codice mostra come utilizzare `stop-flow`.

### AWS CLI

Per interrompere un flusso

L'`stop-flow` esempio seguente interrompe il flusso specificato.

```
aws mediacconnect stop-flow \  
  --flow-arn arn:aws:mediacconnect:us-east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

Output:

```
{  
  "Status": "STOPPING",  
  "FlowArn": "arn:aws:mediacconnect:us-east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow"  
}
```

Per ulteriori informazioni, consulta [Stopping a Flow](#) nella AWS Elemental MediaConnect User Guide.

- Per API i dettagli, consulta AWS CLI Command [StopFlow](#) Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere tag a una MediaConnect risorsa

L'`tag-resource` esempio seguente aggiunge un tag con un nome e un valore di chiave alla MediaConnect risorsa specificata.

```
aws mediaconnect tag-resource \  
  --resource-arn arn:aws:mediaconnect:us-east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame \  
  --tags region=west
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [ListTagsForResource](#), [TagResource](#), [UntagResource](#) nel MediaConnect API Riferimento AWS elementare.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una MediaConnect risorsa

L'`untag-resource` esempio seguente rimuove il tag con il nome di chiave specificato e il relativo valore associato da una MediaConnect risorsa.

```
aws mediaconnect untag-resource \  
  --resource-arn arn:aws:mediaconnect:us-east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame \  
  --tag-keys region
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [ListTagsForResource](#), [TagResource](#), [UntagResource](#) nel MediaConnect API Riferimento AWS elementare.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-flow-entitlement

Il seguente esempio di codice mostra come utilizzare `update-flow-entitlement`.

### AWS CLI

Per aggiornare un diritto

L'`update-flow-entitlement` seguente aggiorna l'autorizzazione specificata con una nuova descrizione e un nuovo sottoscrittore.

```
aws mediaconnect update-flow-entitlement \
  --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
  --entitlement-arn arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement
\
  --description 'For AnyCompany Affiliate' \
  --subscribers 777788889999
```

Output:

```
{
  "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
  "Entitlement": {
    "Name": "AnyCompany_Entitlement",
    "Description": "For AnyCompany Affiliate",
    "EntitlementArn": "arn:aws:mediaconnect: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"
    ]
  }
}
```

```
}
```

Per ulteriori informazioni, consulta [Updating an Entitlement nella Elemental User Guide](#).AWS MediaConnect

- Per API i dettagli, consulta Command [UpdateFlowEntitlement](#)Reference AWS CLI .

## update-flow-output

Il seguente esempio di codice mostra come utilizzare `update-flow-output`.

### AWS CLI

Per aggiornare un output su un flusso

L'`update-flow-output`esempio seguente aggiorna un output sul flusso specificato.

```
aws mediaconnect update-flow-output \
  --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
  --output-arn arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC \
  --port 3331
```

Output:

```
{
  "FlowArn": "arn:aws:mediaconnect: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:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC",
    "Destination": "192.0.2.12"
  }
}
```



Per ulteriori informazioni, vedete [Aggiornamento degli output su un flusso](#) nella Guida per l'MediaConnect utente di AWS Elemental.

- Per API i dettagli, consulta AWS CLI Command [UpdateFlowOutputReference](#).

## update-flow-source

Il seguente esempio di codice mostra come utilizzare `update-flow-source`.

### AWS CLI

Per aggiornare l'origine di un flusso esistente

L'`update-flow-source`esempio seguente aggiorna l'origine di un flusso esistente.

```
aws mediaconnect update-flow-source \  
  --flow-arn arn:aws:mediaconnect:us-  
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow \  
  --source-arn arn:aws:mediaconnect: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
```

Output:

```
{  
  "FlowArn": "arn:aws:mediaconnect: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:mediaconnect:us-  
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource"  
  }  
}
```

Per ulteriori informazioni, consulta [Aggiornamento della fonte di un flusso](#) nella Guida per l' MediaConnect utente di AWS Elemental.

- Per API i dettagli, consulta [UpdateFlowSource AWS CLI](#) Command Reference.

## MediaConvert esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with MediaConvert.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **cancel-job**

Il seguente esempio di codice mostra come utilizzare `cancel-job`.

#### AWS CLI

Per annullare un lavoro in coda

L'`cancel-job` esempio seguente annulla il lavoro con ID. `1234567891234-abc123` Non è possibile annullare un lavoro che il servizio ha iniziato a elaborare.

```
aws mediaconvert cancel-job \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --id 1234567891234-abc123
```

Per ottenere un endpoint specifico per `accountdescribe-endpoints`, usa o invia il comando senza l'endpoint. Il servizio restituisce un errore e il tuo endpoint.

Per ulteriori informazioni, consulta [Lavorare con AWS Elemental MediaConvert Jobs nella Guida per l'utente di AWS MediaConvert Elemental](#).

- Per API i dettagli, consultate AWS CLI Command [CancelJob](#)Reference.

## create-job-template

Il seguente esempio di codice mostra come utilizzare `create-job-template`.

### AWS CLI

Per creare un modello di processo

L'`create-job-template`esempio seguente crea un modello di lavoro con le impostazioni di transcodifica specificate nel file `job-template.json` che si trova sul sistema.

```
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
```

Se create il JSON file del modello di lavoro utilizzando `get-job-template` e quindi modificando il file, rimuovete l'`JobTemplate` oggetto, ma conservate al suo interno l'oggetto secondario `Settings`. Inoltre, assicuratevi di rimuovere le seguenti coppie chiave-valore: `LastUpdated`, `Arn`, `Type` e `CreatedAt`. Puoi specificare la categoria, la descrizione, il nome e la coda nel JSON file o nella riga di comando.

Per ottenere un endpoint specifico per `accountdescribe-endpoints`, usa o invia il comando senza l'endpoint. Il servizio restituisce un errore e il tuo endpoint.

Se la richiesta ha esito positivo, il servizio restituisce le JSON specifiche per il modello di lavoro creato.

Per ulteriori informazioni, consulta [Working with AWS Elemental MediaConvert Job Templates nella Guida per l'utente di AWS Elemental](#).

- Per API i dettagli, consultate AWS CLI Command [CreateJobTemplate](#)Reference.

## create-job

Il seguente esempio di codice mostra come utilizzare `create-job`.

## AWS CLI

Per creare un lavoro

L'`create-job` seguente crea un processo di transcodifica con le impostazioni specificate in un file `job.json` che si trova sul sistema da cui si invia il comando. Questa specifica di JSON lavoro può specificare ogni impostazione singolarmente, fare riferimento a un modello di lavoro o fare riferimento a preimpostazioni di output.

```
aws mediaconvert create-job \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --cli-input-json file://~/job.json
```

Puoi utilizzare la MediaConvert console AWS Elemental per generare le specifiche del JSON lavoro scegliendo le impostazioni del lavoro e quindi selezionando Mostra lavoro nella parte JSON inferiore della sezione Job.

Per ottenere un endpoint specifico per `accountdescribe-endpoints`, usa o invia il comando senza l'endpoint. Il servizio restituisce un errore e il tuo endpoint.

Se la richiesta ha esito positivo, il servizio restituisce le specifiche del JSON lavoro che hai inviato con la richiesta.

Per ulteriori informazioni, consulta [Lavorare con AWS Elemental MediaConvert Jobs nella Guida](#) per l' MediaConvert utente di AWS Elemental.

- Per API i dettagli, consultate AWS CLI Command [CreateJob](#)Reference.

## **create-preset**

Il seguente esempio di codice mostra come utilizzare `create-preset`.

## AWS CLI

Per creare un preset di output personalizzato

L'`create-preset` seguente crea una preimpostazione di output personalizzata in base alle impostazioni di output specificate nel file `preset.json`. È possibile specificare la categoria, la descrizione e il nome nel JSON file o nella riga di comando.

```
aws mediaconvert create-preset \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1 \  
  --cli-input-json file://~/preset.json
```

Se create il JSON file preimpostato utilizzando `get-preset` e poi modificando il file di output, assicuratevi di rimuovere le seguenti coppie chiave-valore: `LastUpdated`, `Arn` e `Type CreatedAt`

Per ottenere un endpoint specifico per l'account, utilizzate `describe-endpoints` o inviate il comando senza l'endpoint. Il servizio restituisce un errore e il tuo endpoint.

Per ulteriori informazioni, consulta [Lavorare con i preset di MediaConvert output AWS Elemental](#) nella Guida per l'utente di AWS MediaConvert Elemental.

- Per API i dettagli, consultate Command [CreatePreset](#) Reference AWS CLI .

## create-queue

Il seguente esempio di codice mostra come utilizzare `create-queue`.

### AWS CLI

Per creare una coda personalizzata

L'`create-queue` esempio seguente crea una coda di transcodifica personalizzata.

```
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."
```

Per ottenere un endpoint specifico per l'account, utilizza o invia il comando senza `describe-endpoints` l'endpoint. Il servizio restituisce un errore e il tuo endpoint.

Output:

```
{  
  "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."
}
```

Per ulteriori informazioni, consulta [Working with AWS Elemental MediaConvert Queues](#) nella Guida per l'utente di AWS MediaConvert Elemental.

- Per API i dettagli, consultate Command [CreateQueue](#) Reference AWS CLI .

## delete-job-template

Il seguente esempio di codice mostra come utilizzare `delete-job-template`.

### AWS CLI

Per eliminare un modello di lavoro

L'`delete-job-template` esempio seguente elimina il modello di lavoro personalizzato specificato.

```
aws mediaconvert delete-job-template \
  --name "DASH Streaming" \
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Questo comando non produce alcun output. Esegui `aws mediaconvert list-job-templates` per confermare che il modello è stato eliminato.

Per ulteriori informazioni, consulta [Working with AWS Elemental MediaConvert Job Templates nella Guida per l'utente di AWS Elemental](#).

- Per API i dettagli, consultate AWS CLI Command [DeleteJobTemplate](#) Reference.

## delete-preset

Il seguente esempio di codice mostra come utilizzare `delete-preset`.

## AWS CLI

Per eliminare una coda personalizzata su richiesta

L'`delete-preset` esempio seguente elimina la preimpostazione personalizzata specificata.

```
aws mediaconvert delete-preset \  
  --name SimpleMP4 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Questo comando non produce alcun output. Esegui `aws mediaconvert list-presets` per confermare che la tua preimpostazione è stata eliminata.

Per ulteriori informazioni, consulta [Lavorare con i preset di MediaConvert output AWS elementali](#) nella Guida per l'utente di AWS MediaConvert Elemental.

- Per API i dettagli, consultate Command [DeletePresetReference](#) AWS CLI .

## `delete-queue`

Il seguente esempio di codice mostra come utilizzare `delete-queue`.

## AWS CLI

Per eliminare una coda personalizzata su richiesta

L'`delete-queue` esempio seguente elimina la coda su richiesta personalizzata specificata.

Non è possibile eliminare la coda predefinita. Non è possibile eliminare una coda riservata che ha un piano tariffario attivo o che contiene processi non elaborati.

```
aws mediaconvert delete-queue \  
  --name Customer1 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Questo comando non produce alcun output. Esegui `aws mediaconvert list-queues` per confermare che la coda è stata eliminata.

Per ulteriori informazioni, consulta [Lavorare con AWS Elemental MediaConvert Queues](#) nella Guida per l'utente di AWS MediaConvert Elemental.

- Per API i dettagli, consultate Command [DeleteQueueReference](#) AWS CLI .

## describe-endpoints

Il seguente esempio di codice mostra come utilizzare `describe-endpoints`.

### AWS CLI

Per ottenere un endpoint specifico per l'account

L'`describe-endpoints` esempio seguente recupera l'endpoint necessario per inviare qualsiasi altra richiesta al servizio.

```
aws mediaconvert describe-endpoints
```

Output:

```
{
  "Endpoints": [
    {
      "Url": "https://abcd1234.mediaconvert.region-name-1.amazonaws.com"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Getting Started with MediaConvert Using the API in the AWS MediaConvert API Elemental Reference](#).

- Per API i dettagli, consulta [DescribeEndpoints AWS CLI Command Reference](#).

## get-job-template

Il seguente esempio di codice mostra come utilizzare `get-job-template`.

### AWS CLI

Per ottenere i dettagli di un modello di lavoro

L'`get-job-template` esempio seguente mostra la JSON definizione del modello di lavoro personalizzato specificato.

```
aws mediaconvert get-job-template \
  --name "DASH Streaming" \
  --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com
```



## Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Working with AWS Elemental MediaConvert Job Templates nella Guida per l'utente di AWS Elemental](#).

- Per API i dettagli, consultate AWS CLI Command [GetJobTemplateReference](#).

## get-job

Il seguente esempio di codice mostra come utilizzare `get-job`.

### AWS CLI

Per ottenere dettagli su un particolare lavoro

L'esempio seguente richiede le informazioni per il lavoro con ID1234567890987-1ab2c3, che in questo esempio è terminato con un errore.

```
aws mediaconvert get-job \
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
  --region region-name-1 \
  --id 1234567890987-1ab2c3
```

Per ottenere l'endpoint specifico dell'`accountdescribe-endpoints`, utilizza o invia il comando senza l'endpoint. Il servizio restituisce un errore e il tuo endpoint.

Se la richiesta ha esito positivo, il servizio restituisce un JSON file con informazioni sul lavoro, incluse le impostazioni del lavoro, eventuali errori restituiti e altri dati sul lavoro, come segue:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Lavorare con AWS Elemental MediaConvert Jobs nella Guida](#) per l' MediaConvert utente di AWS Elemental.

- Per API i dettagli, consultate AWS CLI Command [GetJobReference](#).

## get-preset

Il seguente esempio di codice mostra come utilizzare get-preset.

### AWS CLI

Per ottenere dettagli su un particolare preset

L'get-presetesempio seguente richiede la JSON definizione della preimpostazione personalizzata specificata.

```
aws mediaconvert get-preset \  
  --name SimpleMP4 \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Output:

```
{  
  "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
  }
}

```

Per ulteriori informazioni, vedete [Lavorare con i preset di MediaConvert output AWS Elemental](#) nella Guida per l'utente di AWS MediaConvert Elemental.

- Per API i dettagli, consultate Command [GetPreset](#)Reference AWS CLI .

## get-queue

Il seguente esempio di codice mostra come utilizzare `get-queue`.

### AWS CLI

Per ottenere i dettagli di una coda

L'`get-queue`esempio seguente recupera i dettagli della coda personalizzata specificata.

```

aws mediaconvert get-queue \
  --name Customer1 \
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consultate [Working with AWS Elemental MediaConvert Queues nella Guida](#) per l'utente di Elemental AWS . MediaConvert

- Per API i dettagli, consultate Command [GetQueueReference](#) AWS CLI .

## list-job-templates

Il seguente esempio di codice mostra come utilizzare `list-job-templates`.

### AWS CLI

Esempio 1: per elencare i modelli di lavoro personalizzati

L'`list-job-templates` esempio seguente elenca tutti i modelli di lavoro personalizzati nella regione corrente. Per elencare i modelli di job di sistema, vedete l'esempio successivo.

```
aws mediaconvert list-job-templates \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Output:

```
{  
  "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
    }
]
}

```

Esempio 2: per elencare i modelli MediaConvert di processo di sistema

L'`list-job-templates` seguente elenca tutti i modelli di job di sistema.

```

aws mediaconvert list-job-templates \
  --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com \
  --list-by SYSTEM

```

Output:

```

{
  "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>...  
  ]  
}
```

Per ulteriori informazioni, consulta [Working with AWS Elemental MediaConvert Job Templates nella Guida per l'utente di AWS Elemental](#).

- Per API i dettagli, consultate AWS CLI Command [ListJobTemplates](#)Reference.

## list-jobs

Il seguente esempio di codice mostra come utilizzare `list-jobs`.

### AWS CLI

Per ottenere dettagli su tutti i lavori in una regione

L'esempio seguente richiede le informazioni per tutti i lavori nella regione specificata.

```
aws mediaconvert list-jobs \  
  --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \  
  --region region-name-1
```

Per ottenere un endpoint specifico per `accountdescribe-endpoints`, utilizzate o inviate il comando senza l'endpoint. Il servizio restituisce un errore e il tuo endpoint.

Per ulteriori informazioni, consulta [Lavorare con AWS Elemental MediaConvert Jobs nella Guida per l'utente di AWS MediaConvert Elemental](#).

- Per API i dettagli, consultate AWS CLI Command [ListJobs](#)Reference.

## list-presets

Il seguente esempio di codice mostra come utilizzare `list-presets`.

### AWS CLI

Esempio 1: per elencare i preset di output personalizzati

L'`list-presets` esempio seguente elenca i preset di output personalizzati. Per elencare i preset di sistema, vedete l'esempio successivo.

```
aws mediaconvert list-presets \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Output:

```
{  
  "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."  
    }  
  ]  
}
```

Esempio 2: Per elencare le preimpostazioni di uscita del sistema

L'`list-presets` seguente elenca le preimpostazioni di MediaConvert sistema disponibili. Per elencare i preset personalizzati, consultate l'esempio precedente.

```
aws mediaconvert list-presets \  
  --list-by SYSTEM \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Output:

```

{
  "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, 35mpbs",
      "LastUpdated": 1568321790,
      "Type": "SYSTEM",
      "Category": "MXF",
      "Settings": {
        ...<output settings removed for brevity>...
      }
    }
  ]
}

```

Per ulteriori informazioni, consultate [Lavorare con i preset di MediaConvert output AWS Elemental](#) nella Guida per l'utente di AWS MediaConvert Elemental.

- Per API i dettagli, consultate Command [ListPresets](#) Reference AWS CLI .

## list-queues

Il seguente esempio di codice mostra come utilizzare `list-queues`.

## AWS CLI

Per elencare le code

L'`list-queues` seguente elenca tutte le MediaConvert code.

```
aws mediaconvert list-queues \  
  --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Output:

```
{  
  "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
  }
]
}

```

Per ulteriori informazioni, consulta [Working with AWS Elemental MediaConvert Queues](#) nella Guida per l'utente di AWS MediaConvert Elemental.

- Per API i dettagli, consultate Command [ListQueues](#) Reference AWS CLI .

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag su una MediaConvert coda, un modello di lavoro o una preimpostazione di output

L'`list-tags-for-resource` esempio seguente elenca i tag sulla preimpostazione di output specificata.

```

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

```

Output:

```

{
  "ResourceTags": {
    "Tags": {
      "customer": "zippyVideo"
    }
  }
}

```

```
    },
    "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4"
  }
}
```

Per ulteriori informazioni, consulta [Tagging AWS Elemental MediaConvert Queues, Job Templates e Output Preset](#) nella Guida utente di Elemental AWS . MediaConvert

- Per i API dettagli, consultate Command Reference. [ListTagsForResourceAWS CLI](#)

## update-job-template

Il seguente esempio di codice mostra come utilizzare `update-job-template`.

### AWS CLI

Per modificare un modello di lavoro

L'`update-job-template` esempio seguente sostituisce la JSON definizione del modello di lavoro personalizzato specificato con la JSON definizione nel file fornito.

```
aws mediaconvert update-job-template --name File1 --endpoint-url -- file: //~/ .json https://
abcd1234.mediaconvert.us-west-2.amazonaws.com cli-input-json job-template-update
```

Contenuto di `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
}

```

Il sistema restituisce il JSON payload inviato con la richiesta, anche quando la richiesta genera un errore. Pertanto, la JSON restituita non è necessariamente la nuova definizione del modello di lavoro.

Poiché il JSON payload può essere lungo, potrebbe essere necessario scorrere verso l'alto per visualizzare eventuali messaggi di errore.



Per ulteriori informazioni, consulta [Working with AWS Elemental MediaConvert Job Templates nella Guida per l'utente di AWS Elemental](#).

- Per API i dettagli, consultate AWS CLI Command [UpdateJobTemplateReference](#).

## update-preset

Il seguente esempio di codice mostra come utilizzare `update-preset`.

### AWS CLI

Per modificare un preset

L'`update-preset` seguente sostituisce la descrizione del preset specificato.

```
aws mediaconvert update-preset \  
--name Customer1 \  
--description "New description text."  
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Questo comando non produce alcun output. Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, vedete [Lavorare con i preset di MediaConvert output AWS elementali nella Guida per l'utente di Elemental AWS . MediaConvert](#)

- Per API i dettagli, consultate Command [UpdatePresetReference](#) AWS CLI .

## update-queue

Il seguente esempio di codice mostra come utilizzare `update-queue`.

### AWS CLI

Per modificare una coda

L'`update-queue` esempio seguente mette in pausa la coda specificata, modificandone lo stato in `PAUSED`

```
aws mediaconvert update-queue \  
--name Customer1 \  
--status PAUSED \  
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Working with AWS Elemental MediaConvert Queues nella Guida](#) per l'utente di Elemental AWS . MediaConvert

- Per API i dettagli, consultate Command [UpdateQueue](#) Reference AWS CLI .

## MediaLive esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with MediaLive.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-channel**

Il seguente esempio di codice mostra come utilizzare `create-channel`.

AWS CLI

Per creare un canale

L'`create-channel` esempio seguente crea un canale passando un JSON file che contiene i parametri che desiderate specificare.

Il canale in questo esempio acquisisce un HLS PULL input che si collega a una sorgente che contiene video, audio e didascalie incorporate. Il canale crea un gruppo HLS di output con un server Akamai come destinazione. Il gruppo di output contiene due uscite: una per il video e l'AAC audio H.265 e una per i VTT sottotitoli Web, solo in inglese.

Il canale di questo JSON esempio include i parametri minimi richiesti per un canale che utilizza un HLS PULL input e che produce un gruppo di HLS output con Akamai come destinazione. JSON contiene le seguenti sezioni principali:

`InputAttachments`, che specifica una fonte per l'audio e una fonte per i sottotitoli. Non specifica un selettore video, il che significa che MediaLive estrae il primo video che trova nella sorgente. `Destinations`, che contiene i due indirizzi IP (URLs) per il singolo gruppo di output in questo canale. Questi indirizzi richiedono una password. `EncoderSettings`, che contiene sottosezioni. `AudioDescriptions`, che specifica che il canale contiene una risorsa di uscita audio, che utilizza la sorgente da `InputAttachments` e produce audio in AAC formato. `CaptionDescriptions`, che specifica che il canale contiene una risorsa di output per i sottotitoli, che utilizza la fonte da `InputAttachments` e produce sottotitoli in formato Web. VTT

VideoDescriptions, che specifica che il canale contiene una risorsa di uscita video, con la risoluzione specificata. OutputGroups, che specifica i gruppi di output. In questo esempio esiste un gruppo denominato Akamai. La connessione viene effettuata utilizzando HLSPUT. Il gruppo di output contiene due uscite. Un'uscita è per la risorsa video (denominata Video\_high) e la risorsa audio (denominata Audio\_EN). Un output è per la risorsa didascalie (denominata WebVTT\_EN).

In questo esempio, alcuni parametri non contengono alcun valore o contengono parametri vuoti annidati. Ad esempio, OutputSettings l'Video\_and\_audiooutput contiene diversi parametri annidati che terminano con un parametro vuoto M3U8Settings. Questo parametro deve essere incluso, ma è possibile omettere uno, più o tutti i relativi elementi secondari, il che significa che il parametro secondario assumerà il valore predefinito o sarà nullo.

Tutti i parametri che si applicano a questo canale di esempio ma non sono specificati in questo file assumeranno il valore predefinito, saranno impostati su null o assumeranno un valore univoco generato da MediaLive

```
aws medialive create-channel \  
  --cli-input-json file://channel-in-hls-out-hls-akamai.json
```

Contenuto di 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"
}
}
}

```

### Output:

L'output ripete il contenuto del JSON file, più i seguenti valori. Tutti i parametri sono ordinati alfabeticamente.

ARN per il canale. L'ultima parte di ARN è l'ID univoco del canale. EgressEndpoints è vuoto in questo canale di esempio perché utilizzato solo per gli PUSH input. Quando viene applicato, mostra gli indirizzi a MediaLive cui viene inviato il contenuto. OutputGroups, Outputs. Questi mostrano tutti i parametri per il gruppo di output e le uscite, compresi quelli che non hai incluso ma che sono rilevanti per questo canale. I parametri potrebbero essere vuoti (ad esempio indicando che il parametro o la funzionalità sono disabilitati in questa configurazione del canale) o potrebbero mostrare il valore predefinito che verrà applicato. LogLevel è impostato sul valore predefinito (DISABLED). Tags è impostato sul valore predefinito (null). PipelinesRunningCounte State mostra lo stato attuale del canale.

Per ulteriori informazioni, consulta [Creazione di un canale da zero](#) nella Guida MediaLive utente di AWS Elemental.

- Per API i dettagli, consultate [CreateChannel AWS CLI Command Reference](#).

## create-input

Il seguente esempio di codice mostra come utilizzare `create-input`.

## AWS CLI

Per creare un input

L'`create-input` seguente crea un HLS PULL input passando un JSON file che contiene i parametri che si applicano a questo tipo di input. L'input di questo JSON esempio specifica due fonti (indirizzi) per l'input, in modo da supportare la ridondanza nell'inserimento. Questi indirizzi richiedono password.

```
aws medialive create-input \  
  --cli-input-json file://input-hls-pull-news.json
```

Contenuto di `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"  
}
```

Output:

L'output ripete il contenuto del JSON file, più i seguenti valori. Tutti i parametri sono ordinati alfabeticamente.

`Arn` per l'input. L'ultima parte di ARN è l'ID di input univoco. `Attached Channels`, che è sempre vuoto per un input appena creato. `Destinations`, che in questo esempio è vuoto perché viene utilizzato solo con un PUSH input. `Id` per l'input, lo stesso dell'ID in ARN. `MediaConnectFlows`, che in questo esempio è vuoto perché viene utilizzato solo con un input di tipo MediaConnect.



`SecurityGroups`, che in questo esempio è vuoto perché viene utilizzato solo con un `PUSH` input. `Statedi` questo input. `Tags`, che è vuoto (impostazione predefinita per questo parametro).

Per ulteriori informazioni, consulta [Creazione di un input](#) nella Guida MediaLive utente di AWS Elemental.

- Per API i dettagli, consultate [CreateInput AWS CLI Command Reference](#).

## MediaPackage esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with MediaPackage.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-channel**

Il seguente esempio di codice mostra come utilizzare `create-channel`.

AWS CLI

Per creare un canale

Il `create-channel` comando seguente crea un canale denominato `sportschannel` nell'account corrente.

```
aws mediapackage create-channel --id sportschannel
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Creazione di un canale](#) nella Guida MediaPackage utente di AWS Elemental.

- Per API i dettagli, consultate [CreateChannel AWS CLI Command Reference](#).

## create-origin-endpoint

Il seguente esempio di codice mostra come utilizzare `create-origin-endpoint`.

### AWS CLI

Per creare un endpoint di origine

Il `create-origin-endpoint` comando seguente crea un endpoint di origine denominato `cmaf_sports` con le impostazioni del pacchetto fornite in un JSON file e le impostazioni dell'endpoint specificate.

```
aws mediapackage create-origin-endpoint \  
  --channel-id sportschannel \  
  --id cmaf_sports \  
  --cmaf-package file:///file/path/cmafpkg.json --description "cmaf output of sports" \  
  --id cmaf_sports \  
  --manifest-name sports_channel \  
  --startover-window-seconds 300 \  
  --tags region=west,media=sports \  
  --time-delay-seconds 10
```

Output:

```
{  
  "Arn": "arn:aws:mediapackage:us-west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",  
  "ChannelId": "sportschannel",  
  "CmafPackage": {  
    "HlsManifests": [  
      {  
        "AdMarkers": "PASSTHROUGH",  
        "Id": "cmaf_sports_endpoint",  
        "IncludeIframeOnlyStream": true,  
        "ManifestName": "index",  
        "PlaylistType": "EVENT",  
        "PlaylistWindowSeconds": 300,  
        "ProgramDateTimeIntervalSeconds": 300,  
        "Url": "https://c4af3793bf76b33c.mediapackage.us-west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmaf_sports_endpoint/index.m3u8"  
      }  
    ],  
    "SegmentDurationSeconds": 2,  
    "SegmentPrefix": "sportschannel"  
  },  
  "Description": "cmaf output of sports",  
  "Id": "cmaf_sports",  
  "ManifestName": "sports_channel",  
  "StartoverWindowSeconds": 300,  
}
```

```
"Tags": {
  "region": "west",
  "media": "sports"
},
"TimeDelaySeconds": 10,
"Url": "",
"Whitelist": []
}
```

Per ulteriori informazioni, consulta [Creazione di un endpoint](#) nella Guida per l'utente di AWS MediaPackage Elemental.

- Per API i dettagli, consulta AWS CLI Command [CreateOriginEndpoint](#)Reference.

## delete-channel

Il seguente esempio di codice mostra come utilizzare `delete-channel`.

### AWS CLI

Per eliminare un canale

Il `delete-channel` comando seguente elimina il canale denominato `test`.

```
aws mediapackage delete-channel \
  --id test
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di un canale nella Guida](#) utente di AWS MediaPackage Elemental.

- Per API i dettagli, consultate AWS CLI Command [DeleteChannel](#)Reference.

## delete-origin-endpoint

Il seguente esempio di codice mostra come utilizzare `delete-origin-endpoint`.

### AWS CLI

Per eliminare un endpoint di origine

Il `delete-origin-endpoint` comando seguente elimina l'endpoint di origine denominato `tester2`

```
aws mediapackage delete-origin-endpoint \  
  --id tester2
```

Per ulteriori informazioni, consulta [Eliminazione di un endpoint](#) nella Guida per l'utente di Elemental AWS . MediaPackage

- Per API i dettagli, consulta Command [DeleteOriginEndpointReference AWS CLI](#) .

## describe-channel

Il seguente esempio di codice mostra come utilizzare `describe-channel`.

### AWS CLI

Per descrivere un canale

Il `describe-channel` comando seguente visualizza tutti i dettagli del canale denominato `test`.

```
aws mediapackage describe-channel \  
  --id test
```

Output:

```
{  
  "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": {}
}

```

Per ulteriori informazioni, vedete [Visualizzazione dei dettagli dei canali](https://docs.aws.amazon.com/mediapackage/latest/ug/channels-view.html) < <https://docs.aws.amazon.com/mediapackage/latest/ug/channels-view.html> > nella Guida per l'utente di Elemental AWS MediaPackage

- Per i dettagli, consultate Command Reference. API [DescribeChannel](#) AWS CLI

## describe-origin-endpoint

Il seguente esempio di codice mostra come utilizzare `describe-origin-endpoint`.

### AWS CLI

Per descrivere un endpoint di origine

Il `describe-origin-endpoint` comando seguente visualizza tutti i dettagli dell'endpoint di origine denominato `cmaf_sports`

```
aws mediapackage describe-origin-endpoint \
  --id cmaf_sports
```

Output:

```

{
  "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": []
}
```

Per ulteriori informazioni, consulta [Visualizzare un singolo endpoint](#) nella Guida per l'utente di AWS MediaPackage Elemental.

- Per API i dettagli, consulta AWS CLI Command [DescribeOriginEndpoint](#) Reference.

## list-channels

Il seguente esempio di codice mostra come utilizzare `list-channels`.

### AWS CLI

Per elencare tutti i canali

Il `list-channels` comando seguente elenca tutti i canali configurati nell' AWS account corrente.

```
aws mediapackage list-channels
```

Output:

```

{
  "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": {}
    }
  ]
}

```

Per ulteriori informazioni, consulta [Visualizzazione dei dettagli dei canali](#) nella Guida per l' MediaPackage utente di AWS Elemental.

- Per API i dettagli, consultate [ListChannels AWS CLI](#) Command Reference.

## list-origin-endpoints

Il seguente esempio di codice mostra come utilizzare `list-origin-endpoints`.

### AWS CLI

Per elencare tutti gli endpoint di origine su un canale



Il `list-origin-endpoints` comando seguente elenca tutti gli endpoint di origine configurati sul canale denominato. `test`

```
aws mediapackage list-origin-endpoints \  
  --channel-id test
```

Output:

```
{  
  "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": []
  }
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione di tutti gli endpoint associati a un canale](#) nella Guida per l'utente di AWS MediaPackage Elemental.

- Per API i dettagli, consulta AWS CLI Command [ListOriginEndpoints](#) Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag assegnati a una risorsa

Il `list-tags-for-resource` comando seguente elenca i tag assegnati alla risorsa specificata.

```
aws mediapackage list-tags-for-resource \
```

```
--resource-arn arn:aws:mediapackage:us-west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0
```

Output:

```
{
  "Tags": {
    "region": "west"
  }
}
```

Per ulteriori informazioni, consulta [Tagging Resources in AWS Elemental nella MediaPackage AWS MediaPackage Elemental User Guide](#).

- Per API i dettagli, consulta Command [ListTagsForResource](#) Reference AWS CLI .

## **rotate-ingest-endpoint-credentials**

Il seguente esempio di codice mostra come utilizzare `rotate-ingest-endpoint-credentials`.

AWS CLI

Per ruotare le credenziali di acquisizione

Il `rotate-ingest-endpoint-credentials` comando seguente ruota il DAV nome utente e la password Web per l'endpoint di importazione specificato.

```
aws mediapackage rotate-ingest-endpoint-credentials \  
  --id test \  
  --ingest-endpoint-id 584797f1740548c389a273585dd22a63
```

Output:

```
{
  "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": {}
}

```

Per ulteriori informazioni, consulta [Rotating Credentials on an Input URL nella Guida per l'utente di Elemental.AWS MediaPackage](#)

- Per API i dettagli, consultate Command [RotateIngestEndpointCredentials](#)Reference AWS CLI .

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere un tag a una risorsa

I `tag-resource` comandi seguenti aggiungono una coppia di `region=west` chiavi e valori alla risorsa specificata.

```

aws mediapackage tag-resource \
  --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \
  --tags region=west

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Resources in AWS Elemental nella Guida MediaPackage](#) per l'utente di AWS MediaPackage Elemental.

- Per API i dettagli, consulta Command [TagResource](#)Reference AWS CLI .

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da una risorsa

Il `untag-resource` comando seguente rimuove il tag con la chiave `region` dal canale specificato.

```
aws mediapackage untag-resource \  
  --resource-arn arn:aws:mediapackage:us-west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \  
  --tag-keys region
```

Per ulteriori informazioni, consulta [Tagging Resources in AWS Elemental nella Guida MediaPackage](#) per l'utente di AWS MediaPackage Elemental.

- Per API i dettagli, consulta Command [UntagResource](#)Reference AWS CLI .

## update-channel

Il seguente esempio di codice mostra come utilizzare `update-channel`.

### AWS CLI

Per aggiornare un canale

Il `update-channel` comando seguente aggiorna il canale denominato `sportschannel` per includere la descrizione `24x7 sports`.

```
aws mediapackage update-channel \  
  --id sportschannel \  
  --description "24x7 sports"
```

Output:

```
{
  "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": {}
}
```

Per ulteriori informazioni, consulta [Modifica di un canale](#) nella Guida MediaPackage utente di AWS Elemental.

- Per API i dettagli, consultate [UpdateChannel AWS CLI Command Reference](#).

## update-origin-endpoint

Il seguente esempio di codice mostra come utilizzare `update-origin-endpoint`.

### AWS CLI

Per aggiornare un endpoint di origine

Il `update-origin-endpoint` comando seguente aggiorna l'endpoint di origine denominato `cmaf_sports` Cambia il ritardo in 0 secondi.

```
aws mediapackage update-origin-endpoint \  
  --id cmf_sports \  
  --time-delay-seconds 0
```

Output:

```
{  
  "Arn": "arn:aws:mediapackage:us-  
west-2:111222333:origin_endpoints/1dc6718be36f4f34bb9cd86bc50925e6",  
  "ChannelId": "sportschannel",  
  "CmafPackage": {  
    "HlsManifests": [  
      {  
        "AdMarkers": "NONE",  
        "Id": "cmf_sports_endpoint",  
        "IncludeIframeOnlyStream": false,  
        "PlaylistType": "EVENT",  
        "PlaylistWindowSeconds": 60,  
        "ProgramDateTimeIntervalSeconds": 0,  
        "Url": "https://c4af3793bf76b33c.mediapackage.us-  
west-2.amazonaws.com/out/v1/1dc6718be36f4f34bb9cd86bc50925e6/cmf_sports_endpoint/  
index.m3u8"  
      }  
    ],  
    "SegmentDurationSeconds": 2,  
    "SegmentPrefix": "sportschannel"  
  },  
  "Id": "cmf_sports",  
  "ManifestName": "index",  
  "StartoverWindowSeconds": 0,  
  "Tags": {  
    "region": "west",  
    "media": "sports"  
  },  
  "TimeDelaySeconds": 0,  
  "Url": "",  
  "Whitelist": []  
}
```

Per ulteriori informazioni, consulta [Modifica di un endpoint nella Guida](#) per l' MediaPackage utente di AWS Elemental.

- Per API i dettagli, consulta AWS CLI Command [UpdateOriginEndpointReference](#).

## MediaPackage VOD esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with MediaPackage VOD.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-asset**

Il seguente esempio di codice mostra come utilizzare `create-asset`.

AWS CLI

Per creare una risorsa

L'`create-asset` esempio seguente crea una risorsa denominata `Chicken_Asset` nell'AWS account corrente. La risorsa importa il file `30sec_chicken.smil` in 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
```

Output:

```
{  
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Ingest an Asset](#) nella AWS MediaPackage Elemental User Guide.

- Per API i dettagli, consultate AWS CLI Command [CreateAsset](#) Reference.

## create-packaging-configuration

Il seguente esempio di codice mostra come utilizzare `create-packaging-configuration`.

### AWS CLI

Per creare una configurazione di imballaggio

L'esempio seguente crea una configurazione di imballaggio denominata `new_hls` nel gruppo di pacchetti denominato `hls_chicken`. Questo esempio utilizza un file su disco denominato `hls_pc.json` per fornire i dettagli.

```

aws mediapackage-vod create-packaging-configuration \
  --id new_hls \
  --packaging-group-id hls_chicken \
  --hls-package file://hls_pc.json

```

Contenuto di `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
}
```

#### Output:

```
{
  "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"
        }
      }
    ]
  }
}
```

```
}  
}
```

Per ulteriori informazioni, consulta [Creazione di una configurazione di packaging](#) nella Guida MediaPackage utente di AWS Elemental.

- Per API i dettagli, consultate [CreatePackagingConfiguration AWS CLI](#) Command Reference.

## create-packaging-group

Il seguente esempio di codice mostra come utilizzare `create-packaging-group`.

### AWS CLI

Per creare un gruppo di imballaggi

L'esempio seguente elenca tutti i gruppi di pacchetti configurati nell'AWS account corrente.

```
aws mediapackage-vod create-packaging-group \  
  --id hls_chicken
```

Output:

```
{  
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/  
hls_chicken",  
  "Id": "hls_chicken"  
}
```

Per ulteriori informazioni, consulta [Creazione di un gruppo di pacchetti](#) nella Guida per l'utente di AWS Elemental.

- Per API i dettagli, consultate [CreatePackagingGroup AWS CLI](#) Command Reference.

## delete-asset

Il seguente esempio di codice mostra come utilizzare `delete-asset`.

### AWS CLI

Per eliminare una risorsa

L'`delete-asset` seguente elimina la risorsa denominata `30sec_chicken`.

```
aws mediapackage-vod delete-asset \  
  --id 30sec_chicken
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consultate [Eliminazione di una risorsa](#) nella Guida per l'utente di AWS MediaPackage Elemental.

- Per API i dettagli, consultate AWS CLI Command [DeleteAsset](#) Reference.

## **delete-packaging-configuration**

Il seguente esempio di codice mostra come utilizzare `delete-packaging-configuration`.

AWS CLI

Per eliminare una configurazione di imballaggio

L'`delete-packaging-configuration` seguente elimina la configurazione di imballaggio denominata `CMAF`.

```
aws mediapackage-vod delete-packaging-configuration \  
  --id CMAF
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Eliminazione di una configurazione di packaging](#) nella Guida per l'utente di AWS MediaPackage Elemental.

- Per API i dettagli, consultate AWS CLI Command [DeletePackagingConfiguration](#) Reference.

## **delete-packaging-group**

Il seguente esempio di codice mostra come utilizzare `delete-packaging-group`.

AWS CLI

Per eliminare un gruppo di pacchetti

L'`delete-packaging-group` seguente elimina il gruppo di pacchetti denominato `Dash_widevine`.

```
aws mediapackage-vod delete-packaging-group \  
  --id Dash_widevine
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Eliminazione di un gruppo di pacchetti](#) nella Guida per l'utente di AWS MediaPackage Elemental.

- Per API i dettagli, consultate AWS CLI Command [DeletePackagingGroupReference](#).

## describe-asset

Il seguente esempio di codice mostra come utilizzare `describe-asset`.

### AWS CLI

Per descrivere una risorsa

L'`describe-asset` esempio seguente mostra tutti i dettagli della risorsa denominata `30sec_chicken`.

```
aws mediapackage-vod describe-asset \  
  --id 30sec_chicken
```

Output:

```
{  
  "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"
    }
]
}

```

Per ulteriori informazioni, consultate [Visualizzazione dei dettagli delle risorse](#) nella Guida per l' MediaPackage utente di AWS Elemental.

- Per API i dettagli, consultate [DescribeAsset AWS CLI Command Reference](#).

## describe-packaging-configuration

Il seguente esempio di codice mostra come utilizzare `describe-packaging-configuration`.

### AWS CLI

Per descrivere una configurazione di imballaggio

L'esempio seguente mostra tutti i dettagli della configurazione di imballaggio denominata DASH.

```
aws mediapackage-vod describe-packaging-configuration \
  --id DASH
```

Output:

```

{
  "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"
      }
    }
  ]
}

```

Per ulteriori informazioni, consulta [Visualizzazione dei dettagli della configurazione del pacchetto](#) nella Guida per l' MediaPackage utente di AWS Elemental.

- Per API i dettagli, consultate [DescribePackagingConfiguration AWS CLI Command Reference](#).

## describe-packaging-group

Il seguente esempio di codice mostra come utilizzare `describe-packaging-group`.

### AWS CLI

Per descrivere un gruppo di imballaggi

L'esempio seguente mostra tutti i dettagli del gruppo di imballaggi denominato `Packaging_group_1`.

```

aws mediapackage-vod describe-packaging-group \
  --id Packaging_group_1

```

Output:

```

{
  "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/
Packaging_group_1",
  "Id": "Packaging_group_1"
}

```

Per ulteriori informazioni, consulta [Visualizzazione dei dettagli del gruppo di pacchetti](#) nella Guida per l' MediaPackage utente di AWS Elemental.

- Per API i dettagli, consultate [DescribePackagingGroup AWS CLI Command Reference](#).

## list-assets

Il seguente esempio di codice mostra come utilizzare `list-assets`.

### AWS CLI

Per elencare tutte le risorse

L'`list-assets` esempio seguente elenca tutte le risorse configurate nell' AWS account corrente.

```
aws mediapackage-vod list-assets
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consultate [Visualizzazione dei dettagli degli asset](#) nella Guida per l' MediaPackage utente di AWS Elemental.

- Per API i dettagli, consultate [ListAssets AWS CLI Command Reference](#).

## list-packaging-configurations

Il seguente esempio di codice mostra come utilizzare `list-packaging-configurations`.

### AWS CLI

Per elencare tutte le configurazioni di imballaggio

L'`list-packaging-configurations` esempio seguente elenca tutte le configurazioni di imballaggio configurate nel gruppo di pacchetti denominato. `Packaging_group_1`



```
aws mediapackage-vod list-packaging-configurations \
  --packaging-group-id Packaging_group_1
```

Output:

```
{
  "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://lfgubdvs97.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"
    }
  }
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione dei dettagli della configurazione del pacchetto nella Guida](#) per l' MediaPackage utente di AWS Elemental.

- Per API i dettagli, consultate [ListPackagingConfigurations AWS CLI](#) Command Reference.

## list-packaging-groups

Il seguente esempio di codice mostra come utilizzare `list-packaging-groups`.

### AWS CLI

Per elencare tutti i gruppi di imballaggi

L'`list-packaging-groups` esempio seguente elenca tutti i gruppi di pacchetti configurati nell' AWS account corrente.

```
aws mediapackage-vod list-packaging-groups
```

Output:

```

{
  "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"
    }
]
}
```

Per ulteriori informazioni, consulta [Visualizzazione dei dettagli del gruppo di pacchetti](#) nella Guida per l' MediaPackage utente di AWS Elemental.

- Per API i dettagli, consultate [ListPackagingGroups AWS CLI](#) Command Reference.

## MediaStore Esempi di piani dati che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with MediaStore Data Plane.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **delete-object**

Il seguente esempio di codice mostra come utilizzare `delete-object`.

## AWS CLI

Per eliminare un oggetto

L'`delete-object` esempio seguente elimina l'oggetto specificato.

```
aws mediastore-data delete-object \  
  --endpoint=https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path=/folder_name/README.md
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Eliminazione di un oggetto nella Guida](#) utente di AWS MediaStore Elemental.

- Per API i dettagli, consultate AWS CLI Command [DeleteObject](#) Reference.

## describe-object

Il seguente esempio di codice mostra come utilizzare `describe-object`.

### AWS CLI

Per visualizzare le intestazioni di un oggetto

L'`describe-object` esempio seguente visualizza le intestazioni di un oggetto nel percorso specificato.

```
aws mediastore-data describe-object \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path events/baseball/setup.jpg
```

Output:

```
{  
  "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT",  
  "ContentType": "image/jpeg",  
  "ContentLength": "3860266",  
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9e4dd89ff7f5555555555555555da6d3"  
}
```

Per ulteriori informazioni, vedete [Visualizzazione dei dettagli di un oggetto](#) nella Guida per l'utente di AWS Elemental MediaStore.

- Per API i dettagli, vedere [DescribeObject](#) in AWS CLI Command Reference.

## get-object

Il seguente esempio di codice mostra come utilizzare `get-object`.

### AWS CLI

Esempio 1: scaricare un intero oggetto

L'esempio seguente scarica l'oggetto specificato.

```
aws mediastore-data get-object \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --path events/baseball/setup.jpg setup.jpg
```

Output:

```
{  
  "ContentType": "image/jpeg",  
  "StatusCode": 200,  
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f5555555555555555da6d3",  
  "ContentLength": "3860266",  
  "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT"  
}
```

Esempio 2: per scaricare parte di un oggetto

L'esempio seguente scarica la parte specificata di un oggetto.

```
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"
```

Output:

```
{  
  "StatusCode": 206,  
}
```







L'`put-object` seguente carica un oggetto nel contenitore specificato.

```
aws mediastore-data put-object \  
  --endpoint https://aaabbbccdddee.data.mediastore.us-west-2.amazonaws.com \  
  --body ReadMe.md \  
  --path ReadMe.md \  
  --cache-control "max-age=6, public" \  
  --content-type binary/octet-stream
```

Output:

```
{  
  "ContentSHA256":  
  "f29bc64a9d3732b4b9035125fdb3285f5b6455778edca72414671e0ca3b2e0de",  
  "StorageClass": "TEMPORAL",  
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f55555555555da6d3"  
}
```

Esempio 2: caricare un oggetto in una cartella all'interno di un contenitore

L'`put-object` seguente carica un oggetto nella cartella specificata all'interno di un contenitore.

```
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
```

Output:

```
{  
  "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f55555555555da6d3",  
  "ContentSHA256":  
  "f29bc64a9d3732b4b9035125fdb3285f5b6455778edca72414671e0ca3b2e0de",  
  "StorageClass": "TEMPORAL"  
}
```

Per ulteriori informazioni, consultate [Caricamento di un oggetto](#) nella Guida MediaStore utente di AWS Elemental.

- Per API i dettagli, consultate AWS CLI Command [PutObjectReference](#).

## MediaTailor esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with MediaTailor.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **delete-playback-configuration**

Il seguente esempio di codice mostra come utilizzare `delete-playback-configuration`.

#### AWS CLI

Per eliminare una configurazione

Quanto segue `delete-playback-configuration` elimina una configurazione denominata `campaign_short`.

```
aws mediatailor delete-playback-configuration \  
  --name campaign_short
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di una configurazione](#) nella Guida per l'utente di AWS MediaTailor Elemental.

- Per API i dettagli, vedere [DeletePlaybackConfiguration](#) in AWS CLI Command Reference.

## get-playback-configuration

Il seguente esempio di codice mostra come utilizzare `get-playback-configuration`.

### AWS CLI

Per descrivere una configurazione

Di seguito `get-playback-configuration` vengono visualizzati tutti i dettagli della configurazione denominata `west_campaign`.

```
aws mediatailor get-playback-configuration \  
  --name west_campaign
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Visualizzazione di una configurazione](#) nella Guida per l' MediaTailor utente di AWS Elemental.

- Per API i dettagli, vedere [GetPlaybackConfiguration](#) in AWS CLI Command Reference.

## list-playback-configurations

Il seguente esempio di codice mostra come utilizzare `list-playback-configurations`.

### AWS CLI

Per elencare tutte le configurazioni

Di seguito `list-playback-configurations` vengono visualizzati tutti i dettagli della configurazione sull' AWS account corrente.

```
aws mediatailor list-playback-configurations
```

Output:

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione di una configurazione](https://docs.aws.amazon.com/mediatailor/latest/ug/configurations-view.html) nella Guida per l'utente di Elemental.AWS MediaTailor

- Per i dettagli, consultate [Command Reference. API ListPlaybackConfigurations](#) AWS CLI

## put-playback-configuration

Il seguente esempio di codice mostra come utilizzare `put-playback-configuration`.

### AWS CLI

Per creare una configurazione

Quanto segue `put-playback-configuration` crea una configurazione denominata `campaign_short`.

```
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
```

Output:

```
{  
  "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"
```

```
}
```

Per ulteriori informazioni, consulta [Creazione di una configurazione](#) nella Guida MediaTailor utente di AWS Elemental.

- Per API i dettagli, consultate [PutPlaybackConfiguration AWS CLI](#) Command Reference.

## Esempi di MemoryDB utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with MemoryDB.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **copy-snapshot**

Il seguente esempio di codice mostra come utilizzare `copy-snapshot`.

#### AWS CLI

Per copiare un'istantanea

L'`copy-snapshot` esempio seguente crea una copia di un'istantanea.

```
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
  }
}
}
}

```

Per ulteriori informazioni, vedere [Copiare un'istantanea](#) nella Guida per l'utente di MemoryDB.

- Per i API dettagli, vedere [CopySnapshot](#) in Command Reference.AWS CLI

## create-acl

Il seguente esempio di codice mostra come utilizzare `create-acl`.

### AWS CLI

Per creare un ACL

L'`create-acl` esempio seguente crea un nuovo elenco di controllo degli accessi.

```

aws memorydb create-acl \
  --acl-name "new-acl-1" \
  --user-names "my-user"

```

Output:



```
{
  "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"
  }
}
```

Per ulteriori informazioni, vedere [Autenticazione degli utenti con gli elenchi di controllo degli accessi](#) nella Guida per l'utente di MemoryDB.

- Per i API dettagli, vedere [CreateAcl](#) in Command Reference.AWS CLI

## create-cluster

Il seguente esempio di codice mostra come utilizzare `create-cluster`.

### AWS CLI

Per creare un cluster

L'`create-cluster` esempio seguente crea un nuovo cluster.

```
aws memorydb create-cluster \
  --cluster-name my-new-cluster \
  --node-type db.r6g.large \
  --acl-name my-acl \
  --subnet-group my-sg
```

Output:

```
{
  "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
  }
}

```

Per ulteriori informazioni, vedere [Managing Clusters](#) nella Guida per l'utente di MemoryDB.

- Per API i dettagli, vedere [CreateCluster](#) in AWS CLI Command Reference.

## create-parameter-group

Il seguente esempio di codice mostra come utilizzare `create-parameter-group`.

### AWS CLI

Per creare un gruppo di parametri

L'`create-parameter-group` seguente crea un gruppo di parametri.

```

aws memorydb create-parameter-group \
  --parameter-group-name myRedis6x \
  --family memorydb_redis6 \
  --description "my-parameter-group"

```

Output:

```

{
  "ParameterGroup": {
    "Name": "myredis6x",

```

```
    "Family": "memorydb_redis6",
    "Description": "my-parameter-group",
    "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/myredis6x"
  }
}
```

Per ulteriori informazioni, vedere [Creazione di un gruppo di parametri nella Guida](#) per l'utente di MemoryDB.

- Per API i dettagli, vedere [CreateParameterGroup](#) in AWS CLI Command Reference.

## create-snapshot

Il seguente esempio di codice mostra come utilizzare `create-snapshot`.

### AWS CLI

Per creare un'istantanea

L'`create-snapshot`esempio seguente crea un'istantanea.

```
aws memorydb create-snapshot \  
  --cluster-name my-cluster \  
  --snapshot-name my-cluster-snapshot
```

Output:

```
{
  "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
    }
}
```

Per ulteriori informazioni, vedere [Creazione di istantanee manuali](#) nella Guida per l'utente di MemoryDB.

- Per API i dettagli, vedere [CreateSnapshot](#) in AWS CLI Command Reference.

## create-subnet-group

Il seguente esempio di codice mostra come utilizzare `create-subnet-group`.

### AWS CLI

Per creare un gruppo di sottoreti

L'`create-subnet-group` esempio seguente crea un gruppo di sottoreti.

```
aws memorydb create-subnet-group \
  --subnet-group-name mysubnetgroup \
  --description "my subnet group" \
  --subnet-ids subnet-5623xxxx
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, vedere [Creazione di un gruppo di sottoreti](#) nella Guida per l'utente di MemoryDB.

- Per API i dettagli, vedere [CreateSubnetGroup](#) in AWS CLI Command Reference.

## create-user

Il seguente esempio di codice mostra come utilizzare `create-user`.

### AWS CLI

Per creare un utente

L'`create-user` esempio seguente crea un nuovo utente.

```

aws memorydb create-user \
  --user-name user-name-1 \
  --access-string "~objects:* ~items:* ~public:*" \
  --authentication-mode \
    Passwords="enterapasswordhere",Type=password

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, vedere [Autenticazione degli utenti con gli elenchi di controllo degli accessi](#) nella Guida per l'utente di MemoryDB.

- Per i API dettagli, vedere [CreateUser](#) in Command Reference.AWS CLI

## delete-acl

Il seguente esempio di codice mostra come utilizzare `delete-acl`.

### AWS CLI

Per eliminare un ACL

L'`delete-acl` esempio seguente elimina un elenco di controllo degli accessi.

```
aws memorydb delete-acl \  
  --acl-name "new-acl-1"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Autenticazione degli utenti con gli elenchi di controllo degli accessi](#) nella Guida per l'utente di MemoryDB.

- Per i API dettagli, vedere [DeleteAcl](#) in Command Reference.AWS CLI

## delete-cluster

Il seguente esempio di codice mostra come utilizzare `delete-cluster`.

## AWS CLI

Per eliminare un cluster

L'`delete-cluster` seguente elimina un cluster.

```
aws memorydb delete-cluster \  
  --cluster-name my-new-cluster
```

Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, vedere [Eliminazione di un cluster](#) nella Guida per l'utente di MemoryDB.

- Per API i dettagli, vedere [DeleteCluster](#) in AWS CLI Command Reference.

## `delete-parameter-group`

Il seguente esempio di codice mostra come utilizzare `delete-parameter-group`.

## AWS CLI

Per eliminare un gruppo di parametri

L'`delete-parameter-group` seguente elimina un gruppo di parametri.

```
aws memorydb delete-parameter-group \  
  --parameter-group-name myRedis6x
```

Output:

```
{  
  "ParameterGroup": {  
    "Name": "myredis6x",  
    "Family": "memorydb_redis6",  
    "Description": "my-parameter-group",  
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/myredis6x"  
  }  
}
```

Per ulteriori informazioni, vedere [Eliminazione di un gruppo di parametri](#) nella Guida per l'utente di MemoryDB.

- Per API i dettagli, vedere [DeleteParameterGroup](#) in AWS CLI Command Reference.

## `delete-snapshot`

Il seguente esempio di codice mostra come utilizzare `delete-snapshot`.

## AWS CLI

Per eliminare uno snapshot

L'`delete-snapshot` seguente elimina un'istantanea.

```
aws memorydb delete-snapshot \  
  --snapshot-name my-cluster-snapshot
```

Output:

```
{  
  "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
    }
  }
}

```

Per ulteriori informazioni, vedere [Eliminazione di un'istantanea](#) nella Guida per l'utente di MemoryDB.

- Per i API dettagli, vedere [DeleteSnapshot](#) in Command Reference.AWS CLI

## delete-subnet-group

Il seguente esempio di codice mostra come utilizzare `delete-subnet-group`.

### AWS CLI

Per eliminare un gruppo di sottoreti

L'`delete-subnet-group`esempio seguente elimina una sottorete.

```
aws memorydb delete-subnet-group \
  --subnet-group-name mysubnetgroup
```

Output:

```
{
```

```

    "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"
    }
  }
}

```

Per ulteriori informazioni, vedere [Eliminazione di un gruppo di sottoreti](#) nella Guida per l'utente di MemoryDB.

- Per i API dettagli, vedere [DeleteSubnetGroup](#) in Command Reference.AWS CLI

## delete-user

Il seguente esempio di codice mostra come utilizzare `delete-user`.

### AWS CLI

Per eliminare un utente

L'`delete-user` esempio seguente elimina un utente.

```

aws memorydb delete-user \
  --user-name my-user

```

Output:

```

{
  "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"
  }
}
```

Per ulteriori informazioni, vedere [Autenticazione degli utenti con gli elenchi di controllo degli accessi](#) nella Guida per l'utente di MemoryDB.

- Per i API dettagli, vedere [DeleteUser](#) in Command Reference.AWS CLI

## describe-acls

Il seguente esempio di codice mostra come utilizzare `describe-acls`.

AWS CLI

Per restituire un elenco di ACLs

Il seguente `describe-acls` restituisce un elenco di ACLs

```
aws memorydb describe-acls
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, vedere [Autenticazione degli utenti con le liste di controllo degli accessi nella Guida per l'utente](#) di MemoryDB.

- Per i API dettagli, vedere [DescribeAcls](#) in Command Reference.AWS CLI

## describe-clusters

Il seguente esempio di codice mostra come utilizzare `describe-clusters`.

### AWS CLI

Per restituire un elenco di cluster

Il seguente comando `describe-clusters`` restituisce un elenco di cluster.

```
aws memorydb describe-clusters
```

Output:

```

{
  "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
  }
]
}

```

Per ulteriori informazioni, vedere [Managing clusters](#) nella MemoryDB User Guide.

- Per API i dettagli, vedere [DescribeClusters](#) in AWS CLI Command Reference.

## describe-engine-versions

Il seguente esempio di codice mostra come utilizzare `describe-engine-versions`.

### AWS CLI

Per restituire un elenco di versioni del motore

Quanto segue `describe-engine-versions` restituisce un elenco di versioni del motore.

```
aws memorydb describe-engine-versions
```

Output:

```

{
  "EngineVersions": [
    {
      "EngineVersion": "6.2",

```

```
        "EnginePatchVersion": "6.2.6",
        "ParameterGroupFamily": "memorydb_redis6"
    }
]
}
```

Per ulteriori informazioni, consulta [Versioni e aggiornamenti del motore nella Guida per l'utente di MemoryDB](#).

- Per i API dettagli, vedere [DescribeEngineVersions](#) in Command Reference.AWS CLI

## describe-events

Il seguente esempio di codice mostra come utilizzare `describe-events`.

### AWS CLI

Per restituire un elenco di eventi

Il seguente comando `describe-events` restituisce un elenco di eventi.

```
aws memorydb describe-events
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Monitoring events](#) nella MemoryDB User Guide.

- Per API i dettagli, vedere [DescribeEvents](#) in AWS CLI Command Reference.

## describe-parameter-groups

Il seguente esempio di codice mostra come utilizzare `describe-parameter-groups`.

### AWS CLI

Per restituire un elenco di gruppi di parametri

Il seguente `describe-parameter-groups` restituisce un elenco di gruppi di parametri.

```
aws memorydb describe-parameter-groups
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Configurazione dei parametri del motore utilizzando i gruppi di parametri nella Guida](#) per l'utente di MemoryDB.

- Per API i dettagli, vedere [DescribeParameterGroups](#) in AWS CLI Command Reference.

## describe-parameters

Il seguente esempio di codice mostra come utilizzare `describe-parameters`.

### AWS CLI

Per restituire un elenco di parametri

Il seguente comando `describe-parameters` restituisce un elenco di parametri.

```
aws memorydb describe-parameters
```

Output:

```
{
  "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 policyd",
  "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": "1-1000000000",
        "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"
    }
]
}

```

Per ulteriori informazioni, vedere [Configurazione dei parametri del motore utilizzando gruppi di parametri nella Guida per l'utente di MemoryDB](#).

- Per API i dettagli, vedere [DescribeParameters](#) in AWS CLI Command Reference.

## describe-snapshots

Il seguente esempio di codice mostra come utilizzare `describe-snapshots`.

### AWS CLI

Per restituire un elenco di istantanee

Il seguente comando `describe-snapshots` restituisce un elenco di istantanee.

```
aws memorydb describe-snapshots
```

Output:

```
{
  "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
      }
    }
  ]
}
```

Per ulteriori informazioni, vedere [Snapshot](#) and restore nella Guida per l'utente di MemoryDB.

- Per API i dettagli, vedere [DescribeSnapshots](#) in AWS CLI Command Reference.



## describe-subnet-groups

Il seguente esempio di codice mostra come utilizzare `describe-subnet-groups`.

### AWS CLI

Per restituire un elenco di gruppi di sottoreti

Il seguente `describe-subnet-groups` restituisce un elenco di gruppi di sottoreti.

```
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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Sottoreti e gruppi di sottoreti](#) nella Guida per l'utente di MemoryDB.

- Per i API dettagli, vedere in Command Reference. [DescribeSubnetGroups](#) AWS CLI

## describe-users

Il seguente esempio di codice mostra come utilizzare `describe-users`.

### AWS CLI

Per restituire un elenco di utenti

Il seguente comando `describe-users` restituisce un elenco di utenti.

```
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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Autenticazione degli utenti con le liste di controllo degli accessi nella Guida per l'utente](#) di MemoryDB.

- Per i API dettagli, vedere [DescribeUsers](#) in Command Reference.AWS CLI

## failover-shard

Il seguente esempio di codice mostra come utilizzare `failover-shard`.

### AWS CLI

Failover di uno shard

Il seguente `failover-shard` fallisce su uno shard.

```
aws memorydb failover-shard \  
  --cluster-name my-cluster --shard-name 0001
```

Output:

```
{  
  "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
  }
}

```

Per ulteriori informazioni, vedere [Riduzione al minimo dei tempi di inattività con MultiaZ nella Guida per l'utente di MemoryDB](#).

- Per i API dettagli, vedere in Command Reference. [FailoverShard](#)AWS CLI

## list-allowed-node-type-updates

Il seguente esempio di codice mostra come utilizzare `list-allowed-node-type-updates`.

### AWS CLI

Per restituire un elenco di aggiornamenti dei tipi di nodo consentiti

Il seguente `list-allowed-node-type-updates` restituisce un elenco di aggiornamenti del tipo di nodo disponibili.

```
aws memorydb list-allowed-node-type-updates
```

Output:

```

{
  "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
  }
}

```

Per ulteriori informazioni, vedere [Scaling](#) in the MemoryDB User Guide.

- Per API i dettagli, vedere [ListAllowedNodeTypeUpdates](#) in AWS CLI Command Reference.

## list-tags

Il seguente esempio di codice mostra come utilizzare `list-tags`.

### AWS CLI

Per restituire un elenco di tag

Il seguente `list-tags` restituisce un elenco di tag.

```

aws memorydb list-tags \
  --resource-arn arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster

```

Output:

```

{
  "TagList": [
    {
      "Key": "mytag",
      "Value": "myvalue"
    }
  ]
}

```

```
]
}
```

Per ulteriori informazioni, vedere [Tagging resources](#) nella MemoryDB User Guide.

- Per API i dettagli, vedere [ListTags](#) in AWS CLI Command Reference.

## reset-parameter-group

Il seguente esempio di codice mostra come utilizzare `reset-parameter-group`.

### AWS CLI

Per reimpostare un gruppo di parametri

Quanto segue `reset-parameter-group` reimposta un gruppo di parametri.

```
aws memorydb reset-parameter-group \
  --parameter-group-name my-parameter-group \
  --all-parameters
```

Output:

```
{
  "ParameterGroup": {
    "Name": "my-parameter-group",
    "Family": "memorydb_redis6",
    "Description": "my parameter group",
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/my-parameter-
group"
  }
}
```

Per ulteriori informazioni, vedere [Configurazione dei parametri del motore utilizzando i gruppi di parametri nella Guida](#) per l'utente di MemoryDB.

- Per API i dettagli, vedere [ResetParameterGroup](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

## AWS CLI

Per etichettare una risorsa

Il seguente `tag-resource` aggiunge un tag a una risorsa.

```
aws memorydb tag-resource \  
  --resource-arn arn:aws:memorydb:us-east-1:491658xxxxx:cluster/my-cluster \  
  --tags Key="mykey",Value="myvalue"
```

Output:

```
{  
  "TagList": [  
    {  
      "Key": "mytag",  
      "Value": "myvalue"  
    },  
    {  
      "Key": "mykey",  
      "Value": "myvalue"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Tagging resources](#) nella MemoryDB User Guide.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

## AWS CLI

Per aggiornare un ACL

Il seguente `update-acl` aggiorna un utente ACL aggiungendo un utente.

```
aws memorydb untag-resource \  
  --resource-arn arn:aws:memorydb:us-east-1:491658xxxxx:cluster/my-cluster \  
  --tag-keys mykey
```

**Output:**

```
{
  "TagList": [
    {
      "Key": "mytag",
      "Value": "myvalue"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Tagging resources](#) nella MemoryDB User Guide.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

**update-cluster**

Il seguente esempio di codice mostra come utilizzare `update-cluster`.

**AWS CLI**

Per aggiornare un cluster

Il seguente `update-cluster``` aggiorna il gruppo di parametri di un cluster a. `my-parameter-group`

```
aws memorydb update-cluster \
  --cluster-name my-cluster \
  --parameter-group-name my-parameter-group
```

**Output:**

```
{
  "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
  }
}

```

Per ulteriori informazioni, vedere [Modifica di un cluster](#) nella Guida per l'utente di MemoryDB.

- Per API i dettagli, vedere [UpdateCluster](#) in AWS CLI Command Reference.

## update-parameter-group

Il seguente esempio di codice mostra come utilizzare `update-parameter-group`.

### AWS CLI

Per aggiornare un gruppo di parametri

Il seguente `update-parameter-group` `aggiorna un gruppo di parametri.

```

aws memorydb update-parameter-group \
  --parameter-group-name my-parameter-group \
  --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"

```

Output:

```
{
```

```

    "ParameterGroup": {
      "Name": "my-parameter-group",
      "Family": "memorydb_redis6",
      "Description": "my parameter group",
      "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/my-parameter-
group"
    }
  }
}

```

Per ulteriori informazioni, vedere [Modifica di un gruppo di parametri](#) nella Guida per l'utente di MemoryDB.

- Per API i dettagli, vedere [UpdateParameterGroup](#) in AWS CLI Command Reference.

## update-subnet-group

Il seguente esempio di codice mostra come utilizzare `update-subnet-group`.

### AWS CLI

Per aggiornare un gruppo di sottoreti

Il seguente `update-subnet-group` `aggiorna l'ID di sottorete di un gruppo di sottoreti.

```

aws memorydb update-subnet-group \
  --subnet-group-name my-sg \
  --subnet-ids subnet-01f29d458f3xxxxx

```

Output:

```

{
  "SubnetGroup": {
    "Name": "my-sg-1",
    "Description": "my-sg",
    "VpcId": "vpc-09d2cfc01xxxxxxxx",
    "Subnets": [
      {
        "Identifier": "subnet-01f29d458fxxxxxx",
        "AvailabilityZone": {
          "Name": "us-east-1a"
        }
      }
    ]
  }
}

```

```

    ],
    "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/my-sg"
  }
}

```

Per ulteriori informazioni, vedere [Sottoreti e gruppi di sottoreti nella Guida per l'utente di MemoryDB](#).

- Per i API dettagli, vedere in Command Reference. [UpdateSubnetGroup](#) AWS CLI

## update-user

Il seguente esempio di codice mostra come utilizzare `update-user`.

### AWS CLI

Per aggiornare un utente

Quanto segue `update-user` modifica la stringa di accesso di un utente.

```

aws memorydb update-user \
  --user-name my-user \
  --access-string "off ~objects:* ~items:* ~public:* resetchannels -@all"

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, vedere [Autenticazione degli utenti con gli elenchi di controllo degli accessi](#) nella Guida per l'utente di MemoryDB.

- Per i API dettagli, vedere [UpdateUser](#) in Command Reference.AWS CLI

## MSKEsempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con AmazonMSK.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-cluster**

Il seguente esempio di codice mostra come utilizzare `create-cluster`.

AWS CLI

Per creare un MSK cluster Amazon

L'`create-cluster` esempio seguente crea un MSK cluster denominato `MessagingCluster` con tre nodi broker. Un JSON file denominato `brokernodegroupinfo.json` specifica le tre sottoreti su cui desideri che Amazon MSK distribuisca i nodi del broker. Questo esempio non specifica il livello di monitoraggio, quindi il cluster ottiene il livello. `DEFAULT`

```
aws kafka create-cluster \  
  --cluster-name "MessagingCluster" \  
  --broker-node-group-info file://brokernodegroupinfo.json \  
  --kafka-version "2.2.1" \  
  --number-of-broker-nodes 3
```

## Contenuto di `brokernodegroupinfo.json`.

```
{
  "InstanceType": "kafka.m5.xlarge",
  "BrokerAZDistribution": "DEFAULT",
  "ClientSubnets": [
    "subnet-0123456789111abcd",
    "subnet-0123456789222abcd",
    "subnet-0123456789333abcd"
  ]
}
```

## Output:

```
{
  "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
  "ClusterName": "MessagingCluster",
  "State": "CREATING"
}
```

Per ulteriori informazioni, consulta [Creare un MSK cluster Amazon](#) in Amazon Managed Streaming for Apache Kafka.

- Per API i dettagli, consulta Command [CreateClusterReference](#) AWS CLI .

## **create-configuration**

Il seguente esempio di codice mostra come utilizzare `create-configuration`.

### AWS CLI

Per creare una MSK configurazione Amazon personalizzata

L'`create-configuration` esempio seguente crea una MSK configurazione personalizzata con le proprietà del server specificate nel file di input.

```
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
```

Contenuto di `configuration.txt`.

```
auto.create.topics.enable = true
zookeeper.connection.timeout.ms = 2000
log.roll.ms = 604800000
```

Questo comando non produce alcun output. Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Amazon MSK Configuration Operations](#) nella Amazon Managed Streaming for Apache Kafka Developer Guide.

- Per API i dettagli, consulta Command [CreateConfiguration](#) Reference AWS CLI .

## **describe-cluster**

Il seguente esempio di codice mostra come utilizzare `describe-cluster`.

AWS CLI

Per descrivere un cluster

L'`describe-cluster` esempio seguente descrive un MSK cluster Amazon.

```
aws kafka describe-cluster \
  --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/demo-cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5
```

## Output:

```
{
  "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"
}
}

```

Per ulteriori informazioni, consulta [Listing Amazon MSK Clusters nella Amazon Managed Streaming for Apache Kafka Developer Guide](#).

- Per i API dettagli, consulta [DescribeCluster](#) Command Reference.AWS CLI

## get-bootstrap-brokers

Il seguente esempio di codice mostra come utilizzare `get-bootstrap-brokers`.

### AWS CLI

Per ottenere broker bootstrap

L'`get-bootstrap-brokers` esempio seguente recupera le informazioni del broker bootstrap per un cluster AmazonMSK.

```

aws kafka get-bootstrap-brokers \
  --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5

```

Output:

```

{
  "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"
}
```

Per ulteriori informazioni, consulta [Getting the Bootstrap Brokers](#) nella Amazon Managed Streaming for Apache Kafka Developer Guide.

- Per i API dettagli, consulta Command Reference. [GetBootstrapBrokers](#) AWS CLI

## list-clusters

Il seguente esempio di codice mostra come utilizzare `list-clusters`.

### AWS CLI

Per elencare i cluster disponibili

L'`list-clusters` esempio seguente elenca i MSK cluster Amazon presenti nel tuo AWS account.

```
aws kafka list-clusters
```

Output:

```
{
  "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": {},
    "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Listing Amazon MSK Clusters nella Amazon Managed Streaming for Apache Kafka Developer Guide](#).

- Per i API dettagli, consulta [ListClusters](#) Command Reference.AWS CLI

## update-broker-storage

Il seguente esempio di codice mostra come utilizzare `update-broker-storage`.

### AWS CLI

Per aggiornare lo EBS spazio di archiviazione per i broker

L'`update-broker-storage` esempio seguente aggiorna la quantità di EBS spazio di archiviazione per tutti i broker del cluster. Amazon MSK imposta la quantità di storage di destinazione per ogni broker sulla quantità specificata nell'esempio. Puoi ottenere la versione corrente del cluster descrivendo il cluster o elencando tutti i cluster.

```
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-ebs-volume-info "KafkaBrokerNodeId=ALL,VolumeSizeGB=1100"
```

L'output restituisce un valore ARN per questa `update-broker-storage` operazione. Per determinare se questa operazione è completa, utilizzate il `describe-cluster-operation` comando con this ARN come input.

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Update the EBS Storage for Brokers](#) nella Amazon Managed Streaming for Apache Kafka Developer Guide.

- Per i API dettagli, consulta [UpdateBrokerStorage](#) Command Reference.AWS CLI

## update-cluster-configuration

Il seguente esempio di codice mostra come utilizzare `update-cluster-configuration`.

## AWS CLI

Per aggiornare la configurazione di un MSK cluster Amazon

L'update-cluster-configurationesempio seguente aggiorna la configurazione del MSK cluster esistente specificato. Utilizza una MSK configurazione personalizzata.

```
aws kafka update-cluster-configuration \  
  --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"
```

Contenuto di configuration-info.json.

```
{  
  "Arn": "arn:aws:kafka:us-west-2:123456789012:configuration/CustomConfiguration/  
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",  
  "Revision": 1  
}
```

L'output restituisce un valore ARN per questa update-cluster-configuration operazione. Per determinare se questa operazione è completa, utilizzate il describe-cluster-operation comando con this ARN come input.

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Update the Configuration of an Amazon MSK Cluster nella Amazon Managed Streaming for Apache Kafka Developer Guide](#).

- Per API i dettagli, consulta Command [UpdateClusterConfiguration](#)Reference AWS CLI .

## Esempi di Network Manager che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Network Manager.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **associate-customer-gateway**

Il seguente esempio di codice mostra come utilizzare `associate-customer-gateway`.

### AWS CLI

Per associare un gateway per i clienti

L'esempio seguente associa il gateway del cliente `cgw-11223344556677889` nella rete globale specificata al dispositivo `device-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-01231231231231 \  
  --device-id device-07f6fd08867abc123 \  
  --region us-west-2
```

Output:

```
{  
  "CustomerGatewayAssociation": {
```

```

    "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889",
    "GlobalNetworkId": "global-network-01231231231231231",
    "DeviceId": "device-07f6fd08867abc123",
    "State": "PENDING"
  }
}

```

Per ulteriori informazioni, consulta [Customer Gateway Associations](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [AssociateCustomerGateway](#) in AWS CLI Command Reference.

## associate-link

Il seguente esempio di codice mostra come utilizzare `associate-link`.

### AWS CLI

Per associare un collegamento

L'esempio seguente associa il link `link-11112222aaaabbbb1` al dispositivo `device-07f6fd08867abc123`. Il collegamento e il dispositivo si trovano nella rete globale specificata.

```

aws networkmanager associate-link \
  --global-network-id global-network-01231231231231231 \
  --device-id device-07f6fd08867abc123 \
  --link-id link-11112222aaaabbbb1 \
  --region us-west-2

```

Output:

```

{
  "LinkAssociation": {
    "GlobalNetworkId": "global-network-01231231231231231",
    "DeviceId": "device-07f6fd08867abc123",
    "LinkId": "link-11112222aaaabbbb1",
    "LinkAssociationState": "PENDING"
  }
}

```

Per ulteriori informazioni, vedere [Device and Link Associations](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [AssociateLink](#) in AWS CLI Command Reference.

## create-core-network

Il seguente esempio di codice mostra come utilizzare `create-core-network`.

### AWS CLI

Per creare una rete centrale

L'esempio seguente crea una rete principale utilizzando una descrizione e tag opzionali all'interno di una rete WAN globale AWS Cloud.

```
aws networkmanager create-core-network \  
  --global-network-id global-network-cdef-EXAMPLE22222 \  
  --description "Main headquarters location" \  
  --tags Key=Name,Value="New York City office"
```

Output:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Global and core networks](#) nella AWS Cloud WAN User Guide.

- Per API i dettagli, vedere [CreateCoreNetwork](#) in AWS CLI Command Reference.

## create-device

Il seguente esempio di codice mostra come utilizzare `create-device`.

### AWS CLI

Per creare un dispositivo

L'esempio seguente crea un dispositivo nella rete globale specificata. I dettagli del dispositivo includono una descrizione, il tipo, il fornitore, il modello e il numero di serie.

```
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
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, vedere [Working with Devices](#) nella Transit Gateway Network Manager Guide.



- Per API i dettagli, vedere [CreateDevice](#) in AWS CLI Command Reference.

## create-global-network

Il seguente esempio di codice mostra come utilizzare `create-global-network`.

### AWS CLI

Per creare una rete globale

`create-global-network` Gli esempi seguenti creano una nuova rete globale. Lo stato iniziale al momento della creazione è `PENDING`.

```
aws networkmanager create-global-network
```

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [CreateGlobalNetwork](#) in AWS CLI Command Reference.

## create-link

Il seguente esempio di codice mostra come utilizzare `create-link`.

### AWS CLI

Per creare un collegamento

L'`create-link` esempio seguente crea un collegamento nella rete globale specificata. Il collegamento include una descrizione e dettagli sul tipo di collegamento, sulla larghezza di banda e sul provider. L'ID del sito indica il sito a cui è associato il collegamento.

```
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
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Working with Links](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [CreateLink](#) in AWS CLI Command Reference.

## create-site

Il seguente esempio di codice mostra come utilizzare `create-site`.

### AWS CLI

Per creare un sito

L'`create-site`esempio seguente crea un sito nella rete globale specificata. I dettagli del sito includono una descrizione e informazioni sulla posizione.

```
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
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Working with Sites](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [CreateSite](#) in AWS CLI Command Reference.

## **create-vpc-attachment**

Il seguente esempio di codice mostra come utilizzare `create-vpc-attachment`.

### AWS CLI

Per creare un VPC allegato

L'`create-vpc-attachment`esempio seguente crea un VPC allegato con IPv6 supporto in una rete principale.

```
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
```

Output:

```
{  
  "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  
    }  
  }  
}
```

Per ulteriori informazioni, consulta [Create an attachment](#) in the Cloud WAN User Guide.

- Per API i dettagli, consulta [CreateVpcAttachment AWS CLI Command Reference](#).

## delete-attachment

Il seguente esempio di codice mostra come utilizzare `delete-attachment`.

## AWS CLI

Per eliminare un allegato

L'`delete-attachment` seguente elimina un allegato Connect.

```
aws networkmanager delete-attachment \  
--attachment-id attachment-01feddaee26ab68c
```

Output:

```
{  
  "Attachment": {  
    "CoreNetworkId": "core-network-0f4b0a9d5ee7761d1",  
    "AttachmentId": "attachment-01feddaee26ab68c",  
    "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Eliminare gli allegati](#) nella Cloud WAN User Guide.

- Per API i dettagli, consulta [DeleteAttachment AWS CLI Command Reference](#).

## `delete-bucket-analytics-configuration`

Il seguente esempio di codice mostra come utilizzare `delete-bucket-analytics-configuration`.

## AWS CLI

Per eliminare una configurazione di analisi per un bucket

L'`delete-bucket-analytics-configuration` seguente rimuove la configurazione di analisi per il bucket e l'ID specificati.

```
aws s3api delete-bucket-analytics-configuration \  
  --bucket my-bucket \  
  --id 1
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteBucketAnalyticsConfiguration](#) in AWS CLI Command Reference.

## delete-bucket-metrics-configuration

Il seguente esempio di codice mostra come utilizzare `delete-bucket-metrics-configuration`.

### AWS CLI

Per eliminare una configurazione di metriche per un bucket

L'`delete-bucket-metrics-configuration` esempio seguente rimuove la configurazione delle metriche per il bucket e l'ID specificati.

```
aws s3api delete-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

Questo comando non produce alcun output.

- Per API i dettagli, vedete [DeleteBucketMetricsConfiguration](#) in AWS CLI Command Reference.

## delete-core-network

Il seguente esempio di codice mostra come utilizzare `delete-core-network`.

### AWS CLI

Per eliminare una rete centrale

L'`delete-core-network` esempio seguente elimina una rete principale da una rete WAN globale Cloud.

```
aws networkmanager delete-core-network \  
  --core-network-id core-network-0fab62fe438d94db6
```

Output:

```
{
  "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": []
      }
    ]
  }
}
```

Per ulteriori informazioni, consulta [Core networks](#) in the Cloud WAN User Guide.

- Per API i dettagli, consulta [DeleteCoreNetwork AWS CLI](#) Command Reference.

## delete-device

Il seguente esempio di codice mostra come utilizzare `delete-device`.

### AWS CLI

Per eliminare un dispositivo

L'esempio seguente elimina il dispositivo specificato dalla rete globale specificata.

```
aws networkmanager delete-device \
```

```
--global-network-id global-network-01231231231231231 \  
--device-id device-07f6fd08867abc123 \  
--region us-west-2
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Working with Devices](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [DeleteDevice](#) in AWS CLI Command Reference.

## delete-global-network

Il seguente esempio di codice mostra come utilizzare `delete-global-network`.

AWS CLI

Per eliminare una rete globale

L'`delete-global-network` esempio seguente elimina una rete globale.

```
aws networkmanager delete-global-network \  
--global-network-id global-network-052bedddccb193b6b
```

Output:



```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [DeleteGlobalNetwork](#) in AWS CLI Command Reference.

## delete-link

Il seguente esempio di codice mostra come utilizzare `delete-link`.

### AWS CLI

Per eliminare un collegamento

L'`delete-link` esempio seguente elimina il link specificato dalla rete globale specificata.

```
aws networkmanager delete-link \
  --global-network-id global-network-01231231231231231 \
  --link-id link-11112222aaaabbbb1 \
  --region us-west-2
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, vedere [Working with Links](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [DeleteLink](#) in AWS CLI Command Reference.

## **delete-public-access-block**

Il seguente esempio di codice mostra come utilizzare `delete-public-access-block`.

### AWS CLI

Per eliminare la configurazione di blocco dell'accesso pubblico per un bucket

L'`delete-public-access-block` esempio seguente rimuove la configurazione di accesso pubblico a blocchi sul bucket specificato.

```
aws s3api delete-public-access-block \
  --bucket my-bucket
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeletePublicAccessBlock](#) in AWS CLI Command Reference.

## **delete-site**

Il seguente esempio di codice mostra come utilizzare `delete-site`.

### AWS CLI

Per eliminare un sito

L'`delete-site` esempio seguente elimina il sito specificato (`site-444555aaabbb11223`) nella rete globale specificata.

```
aws networkmanager delete-site \
```

```
--global-network-id global-network-01231231231231231 \  
--site-id site-444555aaabbb11223 \  
--region us-west-2
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Working with Sites](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [DeleteSite](#) in AWS CLI Command Reference.

## deregister-transit-gateway

Il seguente esempio di codice mostra come utilizzare `deregister-transit-gateway`.

### AWS CLI

Per annullare la registrazione di un gateway di transito da una rete globale

L'`deregister-transit-gateway` seguente annulla la registrazione del gateway di transito specificato dalla rete globale specificata.

```
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
```

Output:

```
{
  "TransitGatewayRegistration": {
    "GlobalNetworkId": "global-network-01231231231231231",
    "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc",
    "State": {
      "Code": "DELETING"
    }
  }
}
```

Per ulteriori informazioni, vedere [Transit Gateway Registrations](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [DeregisterTransitGateway](#) in AWS CLI Command Reference.

## describe-global-networks

Il seguente esempio di codice mostra come utilizzare `describe-global-networks`.

AWS CLI

Per descrivere le vostre reti globali

L'`describe-global-networks`esempio seguente descrive tutte le reti globali presenti nel tuo account.

```
aws networkmanager describe-global-networks \
  --region us-west-2
```

Output:

```
{
  "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"
    }
]
}

```

- Per API i dettagli, vedere [DescribeGlobalNetworks](#) in AWS CLI Command Reference.

## disassociate-customer-gateway

Il seguente esempio di codice mostra come utilizzare `disassociate-customer-gateway`.

### AWS CLI

Dissociare un Customer Gateway

L'`disassociate-customer-gateway` esempio seguente dissocia il customer gateway specificato (`cgw-11223344556677889`) dalla rete globale specificata.

```

aws networkmanager disassociate-customer-gateway \
  --global-network-id global-network-01231231231231231 \
  --customer-gateway-arn arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889 \
  --region us-west-2

```

Output:

```

{
  "CustomerGatewayAssociation": {
    "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889",
    "GlobalNetworkId": "global-network-01231231231231231",
    "DeviceId": "device-07f6fd08867abc123",
    "State": "DELETING"
  }
}

```

Per ulteriori informazioni, consulta [Customer Gateway Associations](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [DisassociateCustomerGateway](#) in AWS CLI Command Reference.

## disassociate-link

Il seguente esempio di codice mostra come utilizzare `disassociate-link`.

### AWS CLI

Per dissociare un collegamento

L'esempio seguente dissocia il collegamento specificato dal dispositivo `device-07f6fd08867abc123` nella rete globale specificata.

```
aws networkmanager disassociate-link \
  --global-network-id global-network-01231231231231231 \
  --device-id device-07f6fd08867abc123 \
  --link-id link-11112222aaaabbbb1 \
  --region us-west-2
```

Output:

```
{
  "LinkAssociation": {
    "GlobalNetworkId": "global-network-01231231231231231",
    "DeviceId": "device-07f6fd08867abc123",
    "LinkId": "link-11112222aaaabbbb1",
    "LinkAssociationState": "DELETING"
  }
}
```

Per ulteriori informazioni, vedere [Device and Link Associations](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [DisassociateLink](#) in AWS CLI Command Reference.

## get-bucket-analytics-configuration

Il seguente esempio di codice mostra come utilizzare `get-bucket-analytics-configuration`.

### AWS CLI

Per recuperare la configurazione di analisi per un bucket con un ID specifico

L'`get-bucket-analytics-configuration` esempio seguente mostra la configurazione di analisi per il bucket e l'ID specificati.

```
aws s3api get-bucket-analytics-configuration \  
  --bucket my-bucket \  
  --id 1
```

Output:

```
{  
  "AnalyticsConfiguration": {  
    "StorageClassAnalysis": {},  
    "Id": "1"  
  }  
}
```

- Per API i dettagli, vedere [GetBucketAnalyticsConfiguration](#) in AWS CLI Command Reference.

## **get-bucket-metrics-configuration**

Il seguente esempio di codice mostra come utilizzare `get-bucket-metrics-configuration`.

AWS CLI

Per recuperare la configurazione delle metriche per un bucket con un ID specifico

L'`get-bucket-metrics-configuration` esempio seguente visualizza la configurazione delle metriche per il bucket e l'ID specificati.

```
aws s3api get-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

Output:

```
{  
  "MetricsConfiguration": {  
    "Filter": {  
      "Prefix": "logs"  
    },  
  },  
}
```

```

    "Id": "123"
  }
}

```

- Per API i dettagli, vedete [GetBucketMetricsConfiguration](#) in AWS CLI Command Reference.

## get-customer-gateway-associations

Il seguente esempio di codice mostra come utilizzare `get-customer-gateway-associations`.

### AWS CLI

Per ottenere le associazioni dei vostri customer gateway

L'`get-customer-gateway-associations` seguente ottiene le associazioni dei gateway dei clienti per la rete globale specificata.

```

aws networkmanager get-customer-gateway-associations \
  --global-network-id global-network-01231231231231 \
  --region us-west-2

```

Output:

```

{
  "CustomerGatewayAssociations": [
    {
      "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/cgw-11223344556677889",
      "GlobalNetworkId": "global-network-01231231231231231",
      "DeviceId": "device-07f6fd08867abc123",
      "State": "AVAILABLE"
    }
  ]
}

```

- Per API i dettagli, vedere [GetCustomerGatewayAssociations](#) in AWS CLI Command Reference.

## get-devices

Il seguente esempio di codice mostra come utilizzare `get-devices`.



## AWS CLI

Per procurarti i tuoi dispositivi

L'`get-devices`esempio seguente ottiene i dispositivi nella rete globale specificata.

```
aws networkmanager get-devices \  
  --global-network-id global-network-01231231231231231 \  
  --region us-west-2
```

Output:

```
{  
  "Devices": [  
    {  
      "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",  
      "CreatedAt": 1575554005.0,  
      "State": "AVAILABLE"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [GetDevices](#)in AWS CLI Command Reference.

## `get-link-associations`

Il seguente esempio di codice mostra come utilizzare `get-link-associations`.

## AWS CLI

Per ottenere le associazioni dei link

L'`get-link-associations`esempio seguente ottiene le associazioni di link nella rete globale specificata.

```
aws networkmanager get-link-associations \  
  --global-network-id global-network-01231231231231231 \  
  --region us-west-2
```

Output:

```
{  
  "LinkAssociations": [  
    {  
      "GlobalNetworkId": "global-network-01231231231231231",  
      "DeviceId": "device-07f6fd08867abc123",  
      "LinkId": "link-11112222aaaabbbb1",  
      "LinkAssociationState": "AVAILABLE"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [GetLinkAssociations](#) in AWS CLI Command Reference.

## get-links

Il seguente esempio di codice mostra come utilizzare `get-links`.

### AWS CLI

Per ottenere i tuoi link

L'`get-links` esempio seguente ottiene i collegamenti nella rete globale specificata.

```
aws networkmanager get-links \  
  --global-network-id global-network-01231231231231231 \  
  --region us-west-2
```

Output:

```
{  
  "Links": [  
    {  
      "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": "AVAILABLE"
  }
]
}

```

- Per API i dettagli, vedere [GetLinks](#) in AWS CLI Command Reference.

## get-object-retention

Il seguente esempio di codice mostra come utilizzare `get-object-retention`.

### AWS CLI

Per recuperare la configurazione di conservazione degli oggetti per un oggetto

L'`get-object-retention` esempio seguente recupera la configurazione di conservazione degli oggetti per l'oggetto specificato.

```

aws s3api get-object-retention \
  --bucket my-bucket-with-object-lock \
  --key doc1.rtf

```

Output:

```

{
  "Retention": {
    "Mode": "GOVERNANCE",
    "RetainUntilDate": "2025-01-01T00:00:00.000Z"
  }
}

```

- Per API i dettagli, vedere [GetObjectRetention](#) in AWS CLI Command Reference.

## get-public-access-block

Il seguente esempio di codice mostra come utilizzare `get-public-access-block`.

### AWS CLI

Per impostare o modificare la configurazione di accesso pubblico a blocchi per un bucket

L'`get-public-access-block` esempio seguente visualizza la configurazione dell'accesso pubblico a blocchi per il bucket specificato.

```
aws s3api get-public-access-block --bucket my-bucket
```

Output:

```
{
  "PublicAccessBlockConfiguration": {
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "BlockPublicAcls": true,
    "RestrictPublicBuckets": true
  }
}
```

- Per API i dettagli, vedere [GetPublicAccessBlock](#) in AWS CLI Command Reference.

## get-sites

Il seguente esempio di codice mostra come utilizzare `get-sites`.

### AWS CLI

Per accedere ai tuoi siti

L'`get-sites` esempio seguente ottiene i siti della rete globale specificata.

```
aws networkmanager get-sites \
  --global-network-id global-network-01231231231231231 \
  --region us-west-2
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [GetSites](#) in AWS CLI Command Reference.

## get-transit-gateway-registrations

Il seguente esempio di codice mostra come utilizzare `get-transit-gateway-registrations`.

### AWS CLI

Per ottenere le registrazioni dei gateway di transito

L'`get-transit-gateway-registrations`esempio seguente ottiene i gateway di transito registrati nella rete globale specificata.

```
aws networkmanager get-transit-gateway-registrations \
  --global-network-id global-network-01231231231231231 \
  --region us-west-2
```

Output:

```
{
  "TransitGatewayRegistrations": [
    {
      "GlobalNetworkId": "global-network-01231231231231231",
```

```

        "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-
gateway/tgw-123abc05e04123abc",
        "State": {
            "Code": "AVAILABLE"
        }
    }
]
}

```

- Per API i dettagli, vedere [GetTransitGatewayRegistrations](#) in AWS CLI Command Reference.

## get-vpc-attachment

Il seguente esempio di codice mostra come utilizzare `get-vpc-attachment`.

### AWS CLI

Per ottenere un VPC allegato

L'`get-vpc-attachment` esempio seguente restituisce informazioni su un VPC allegato.

```

aws networkmanager get-vpc-attachment \
  --attachment-id attachment-03b7ea450134787da

```

Output:

```

{
  "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
  }
}

```

Per ulteriori informazioni, consulta la Guida per l'WANutente [Attachments](#) in the Cloud.

- Per API i dettagli, consulta [GetVpcAttachment AWS CLI Command Reference](#).

## list-bucket-analytics-configurations

Il seguente esempio di codice mostra come utilizzare `list-bucket-analytics-configurations`.

### AWS CLI

Per recuperare un elenco di configurazioni di analisi per un bucket

Quanto segue `list-bucket-analytics-configurations` recupera un elenco di configurazioni di analisi per il bucket specificato.

```
aws s3api list-bucket-analytics-configurations \
  --bucket my-bucket
```

Output:

```
{
  "AnalyticsConfigurationList": [
    {
      "StorageClassAnalysis": {},
      "Id": "1"
    }
  ]
}
```

```
  ],
  "IsTruncated": false
}
```

- Per API i dettagli, vedere [ListBucketAnalyticsConfigurations](#) in AWS CLI Command Reference.

## list-bucket-metrics-configurations

Il seguente esempio di codice mostra come utilizzare `list-bucket-metrics-configurations`.

### AWS CLI

Per recuperare un elenco di configurazioni di metriche per un bucket

L'`list-bucket-metrics-configurations` esempio seguente recupera un elenco di configurazioni di metriche per il bucket specificato.

```
aws s3api list-bucket-metrics-configurations \
  --bucket my-bucket
```

Output:

```
{
  "IsTruncated": false,
  "MetricsConfigurationList": [
    {
      "Filter": {
        "Prefix": "logs"
      },
      "Id": "123"
    },
    {
      "Filter": {
        "Prefix": "tmp"
      },
      "Id": "234"
    }
  ]
}
```

- Per i API dettagli, vedete [ListBucketMetricsConfigurations](#) in Command Reference.AWS CLI



## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una risorsa

L'`list-tags-for-resource` esempio seguente elenca i tag per la risorsa del dispositivo specificata (`device-07f6fd08867abc123`).

```
aws networkmanager list-tags-for-resource \
  --resource-arn arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231231/device-07f6fd08867abc123 \
  --region us-west-2
```

Output:

```
{
  "TagList": [
    {
      "Key": "Network",
      "Value": "Northeast"
    }
  ]
}
```

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## put-bucket-metrics-configuration

Il seguente esempio di codice mostra come utilizzare `put-bucket-metrics-configuration`.

### AWS CLI

Per impostare una configurazione delle metriche per un bucket

L'`put-bucket-metrics-configuration` esempio seguente imposta una configurazione metrica con ID 123 per il bucket specificato.

```
aws s3api put-bucket-metrics-configuration \
```

```
--bucket my-bucket \  
--id 123 \  
--metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutBucketMetricsConfiguration](#) in AWS CLI Command Reference.

## put-object-retention

Il seguente esempio di codice mostra come utilizzare `put-object-retention`.

### AWS CLI

Per impostare una configurazione di conservazione degli oggetti per un oggetto

L'`put-object-retention` esempio seguente imposta una configurazione di conservazione degli oggetti per l'oggetto specificato fino al 01/01/2025.

```
aws s3api put-object-retention \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf \  
  --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Questo comando non produce alcun output.

- Per i API dettagli, vedere in Command Reference. [PutObjectRetention](#) AWS CLI

## put-public-access-block

Il seguente esempio di codice mostra come utilizzare `put-public-access-block`.

### AWS CLI

Per impostare la configurazione di accesso pubblico a blocchi per un bucket

L'`put-public-access-block` esempio seguente imposta una configurazione restrittiva di accesso pubblico a blocchi per il bucket specificato.

```
aws s3api put-public-access-block \  
  --bucket my-bucket \  
  --
```

```
--public-access-block-configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutPublicAccessBlock](#) in AWS CLI Command Reference.

## register-transit-gateway

Il seguente esempio di codice mostra come utilizzare `register-transit-gateway`.

### AWS CLI

Per registrare un gateway di transito in una rete globale

L'esempio seguente registra il gateway di transito `tgw-123abc05e04123abc` nella rete globale specificata.

```
aws networkmanager register-transit-gateway \
  --global-network-id global-network-01231231231231 \
  --transit-gateway-arn arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc \
  --region us-west-2
```

Output:

```
{
  "TransitGatewayRegistration": {
    "GlobalNetworkId": "global-network-01231231231231",
    "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc",
    "State": {
      "Code": "PENDING"
    }
  }
}
```

Per ulteriori informazioni, vedere [Transit Gateway Registrations](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [RegisterTransitGateway](#) in AWS CLI Command Reference.

## reject-attachment

Il seguente esempio di codice mostra come utilizzare `reject-attachment`.

### AWS CLI

Rifiutare un allegato

L'`reject-attachment` seguente rifiuta una richiesta di VPC allegato.

```
aws networkmanager reject-attachment \  
  --attachment-id attachment-03b7ea450134787da
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta la sezione [Accettazione degli allegati](#) nella Cloud WAN User Guide.

- Per API i dettagli, consulta [RejectAttachment AWS CLI Command Reference](#).

## start-route-analysis

Il seguente esempio di codice mostra come utilizzare `start-route-analysis`.

### AWS CLI

Per iniziare l'analisi del percorso

L'`start-route-analysis` seguente avvia l'analisi tra un'origine e una destinazione, inclusa l'opzione `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
```

Output:

```
{
  "RouteAnalysis": {
    "GlobalNetworkId": "global-network-00aa0aaa0b0aaa000",
    "OwnerAccountId": "111122223333",
    "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
  }
}
```

Per ulteriori informazioni, vedere [Route Analyzer nella Guida](#) per l'utente di AWS Global Networks for Transit Gateways.

- Per API i dettagli, vedere [StartRouteAnalysis](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per applicare tag a una risorsa

L'`tag-resource` esempio seguente applica il tag `Network=Northeast` al dispositivo `device-07f6fd08867abc123`.

```
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
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una risorsa

L'`untag-resource` esempio seguente rimuove il tag con la chiave `Network` dal dispositivo `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
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-device

Il seguente esempio di codice mostra come utilizzare `update-device`.

### AWS CLI

Per aggiornare un dispositivo

L'esempio seguente aggiorna il dispositivo `device-07f6fd08867abc123` specificando un ID del sito per il dispositivo.

```
aws networkmanager update-device \  
  --global-network-id global-network-01231231231231231 \  
  --device-id device-07f6fd08867abc123 \  
  --site-id site-444555aaabbb11223 \  
  --region us-west-2
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Working with Devices](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [UpdateDevice](#) in AWS CLI Command Reference.

## update-global-network

Il seguente esempio di codice mostra come utilizzare `update-global-network`.

### AWS CLI

Per aggiornare una rete globale

L'esempio seguente aggiorna la descrizione della rete globale `global-network-01231231231231231`.

```
aws networkmanager update-global-network \
  --global-network-id global-network-01231231231231231 \
  --description "Head offices" \
  --region us-west-2
```

Output:

```
{
  "GlobalNetwork": {
    "GlobalNetworkId": "global-network-01231231231231231",
    "GlobalNetworkArn": "arn:aws:networkmanager::123456789012:global-network/global-network-01231231231231231",
    "Description": "Head offices",
    "CreatedAt": 1575553525.0,
    "State": "UPDATING"
  }
}
```

Per ulteriori informazioni, vedere [Global Networks](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [UpdateGlobalNetwork](#) in AWS CLI Command Reference.

## update-link

Il seguente esempio di codice mostra come utilizzare `update-link`.

### AWS CLI

Per aggiornare un collegamento

L'esempio seguente aggiorna le informazioni sulla larghezza di banda per il collegamento `link-11112222aaaabbbb1`.



```
aws networkmanager update-link \  
  --global-network-id global-network-01231231231231231 \  
  --link-id Link-11112222aaaabbbb1 \  
  --bandwidth UploadSpeed=20,DownloadSpeed=20 \  
  --region us-west-2
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Working with Links](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [UpdateLink](#) in AWS CLI Command Reference.

## update-site

Il seguente esempio di codice mostra come utilizzare `update-site`.

### AWS CLI

Per aggiornare un sito

L'update-site esempio seguente aggiorna la descrizione del sito `site-444555aaabbb11223` nella rete globale specificata.

```
aws networkmanager update-site \  
  --global-network-id global-network-01231231231231231 \  
  --site-id site-444555aaabbb11223 \  
  --description "New York Office site" \  
  --region us-west-2
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Working with Sites](#) nella Transit Gateway Network Manager Guide.

- Per API i dettagli, vedere [UpdateSite](#) in AWS CLI Command Reference.

## Esempi di utilizzo di Nimble Studio AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface con Nimble Studio.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **get-eula**

Il seguente esempio di codice mostra come utilizzare `get-eula`.

#### AWS CLI

Per ottenere informazioni sul tuo studio

L'`get-eula` esempio seguente elenca le informazioni su un'EULA.

```
aws nimble get-eula \  
  --eula-id "EULAid"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Accetta la versione EULA](#) nella Guida per l'utente di Amazon Nimble Studio.

- Per API i dettagli, consulta [GetEula AWS CLI Command Reference](#).

### **get-launch-profile-details**

Il seguente esempio di codice mostra come utilizzare `get-launch-profile-details`.

#### AWS CLI

Per elencare i widget disponibili

L'get-launch-profile-detailsempio seguente elenca i dettagli su un profilo di lancio.

```
aws nimble get-launch-profile-details \  
  --studio-id "StudioID" \  
  --launch-profile-id "LaunchProfileID"
```

Output:

```
{  
  "launchProfile": {  
    "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/  
yeG71DwNQEiwNTRT7DrV7Q",  
    "createdAt": "2022-01-27T21:18:59+00:00",  
    "createdBy": "AR0A3002NEHCCYRNDDIFT: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": {
    "resourceArn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
yeG7lDwNQEiwNTRT7DrV7Q"
  },
  "updatedAt": "2022-01-27T21:19:13+00:00",
  "updatedBy": "AR0A3002NEHCCYRNDDIFT: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",
        "ggK2eIw6RQyt8PIee0lD3g",
        "a-D9Wc0VQCKUfxAinCDxaw",
        "RvoNmVXiSrS4LhLTb6ybkw",
        "wtp85BcSTa2NZeNRnMKdjw",
        "Rl-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"
    }
]
}

```

Per ulteriori informazioni, consulta [Creazione di profili di lancio](#) nella Guida per l'utente di Amazon Nimble Studio.

- Per API i dettagli, consulta [GetLaunchProfileDetails AWS CLI Command Reference](#).

## get-launch-profile

Il seguente esempio di codice mostra come utilizzare `get-launch-profile`.

### AWS CLI

Per elencare i widget disponibili

L'`get-launch-profile` seguente elenca le informazioni su un profilo di lancio.

```
aws nimble get-launch-profile \
```

```
--studio-id "StudioID" \  
--launch-profile-id "LaunchProfileID"
```

## Output:

```
{  
  "launchProfile": {  
    "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/  
yeG7lDwNQEiwNTRT7DrV7Q",  
    "createdAt": "2022-01-27T21:18:59+00:00",  
    "createdBy": "ARO3002NEHCCYRNDIFT: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"
  }
]
}
```

Per ulteriori informazioni, consulta [Creazione di profili di lancio](#) nella Guida per l'utente di Amazon Nimble Studio.

- Per API i dettagli, consulta [GetLaunchProfile AWS CLI](#) Command Reference.

## get-studio

Il seguente esempio di codice mostra come utilizzare `get-studio`.

## AWS CLI

Per ottenere informazioni sul tuo studio

L'`get-studio` seguente elenca gli studi presenti nel tuo AWS account.

```
aws nimble get-studio \  
  --studio-id "StudioID"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Cos'è Amazon Nimble Studio?](#) nella Guida per l'utente di Amazon Nimble Studio.

- Per API i dettagli, consulta [GetStudio AWS CLI Command Reference](#).

## **list-eula-acceptances**

Il seguente esempio di codice mostra come utilizzare `list-eula-acceptances`.

## AWS CLI

Per elencare i widget disponibili

L'`list-eula-acceptances` seguente elenca quelli accettati EULAs nel tuo AWS account.

```
aws nimble list-eula-acceptances \  
  --studio-id "StudioID"
```

Output:

```
{  
  "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": "Zq9KNEQPRMWJ7FolSoQgUA",
        "eulaId": "wtp85BcSTa2NZeNRnMKdjwt"
    }
]
}

```

Per ulteriori informazioni, consulta [Accetta la versione EULA](#) nella Guida per l'utente di Amazon Nimble Studio.

- Per API i dettagli, consulta [ListEulaAcceptances AWS CLI Command Reference](#).

## list-eulas

Il seguente esempio di codice mostra come utilizzare `list-eulas`.

### AWS CLI

Per elencare i widget disponibili

L'`list-eulas` esempio seguente li elenca EULAs nel tuo AWS account.

```
aws nimble list-eulas
```

Output:

```

{
  "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": "Rl-J0fM5Sl2hyIiwWIV6hw",
    "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Accetta la versione EULA](#) nella Guida per l'utente di Amazon Nimble Studio.

- Per API i dettagli, consulta [ListEulas AWS CLI Command Reference](#).

## list-launch-profiles

Il seguente esempio di codice mostra come utilizzare `list-launch-profiles`.

### AWS CLI

Per elencare i widget disponibili

L'`list-launch-profiles` esempio seguente elenca i profili di lancio del tuo AWS account.

```
aws nimble list-launch-profiles \
  --studio-id "StudioID"
```

Output:

```

{
  "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": "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-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": "AROA3002NEHCCYRNDDIFT: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"
    }
  ]
}
```

```
}
```

Per ulteriori informazioni, consulta [Creazione di profili di lancio](#) nella Guida per l'utente di Amazon Nimble Studio.

- Per API i dettagli, consulta [ListLaunchProfiles AWS CLI Command Reference](#).

## list-studio-components

Il seguente esempio di codice mostra come utilizzare `list-studio-components`.

### AWS CLI

Per elencare i widget disponibili

L'`list-studio-components` esempio seguente elenca i componenti di studio presenti nel tuo AWS account.

```
aws nimble list-studio-components \
  --studio-id "StudioID"
```

Output:

```
{
  "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": "AR0A3002NEHCCYRND DIFT:i-EXAMPLE11111"
  },
  ...
}

```

Per ulteriori informazioni, consulta [How StudioBuilder works with Amazon Nimble Studio nella Guida](#) per l'utente di Amazon Nimble Studio.

- Per API i dettagli, consulta AWS CLI Command [ListStudioComponents](#)Reference.

## list-studio-members

Il seguente esempio di codice mostra come utilizzare `list-studio-members`.

### AWS CLI

Per elencare i widget disponibili

L'`list-studio-members`esempio seguente elenca i membri dello studio disponibili nel tuo AWS account.

```
aws nimble list-studio-members \
  --studio-id "StudioID"
```

Output:

```
{
  "members": [
    {
      "identityStoreId": "d-EXAMPLE11111",
      "persona": "ADMINISTRATOR",
      "principalId": "EXAMPLE11111-e9fd012a-94ad-4f16-9866-c69a63ab6486"
    }
  ]
}
```

```
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Aggiungere utenti in studio](#) nella Guida per l'utente di Amazon Nimble Studio.

- Per API i dettagli, consulta [ListStudioMembers AWS CLI Command Reference](#).

## list-studios

Il seguente esempio di codice mostra come utilizzare `list-studios`.

### AWS CLI

Per elencare i tuoi studi

L'`list-studios` seguente elenca gli studi presenti nel tuo AWS account.

```
aws nimble list-studios
```

Output:

```
{  
  "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"
    }
]
}
```

Per ulteriori informazioni, consulta [Cos'è Amazon Nimble Studio?](#) nella Guida per l'utente di Amazon Nimble Studio.

- Per API i dettagli, consulta [ListStudios AWS CLI](#) Command Reference.

## OpenSearch Esempi di servizi che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with OpenSearch Service.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **create-elasticsearch-domain**

Il seguente esempio di codice mostra come utilizzare `create-elasticsearch-domain`.

#### AWS CLI

Per creare un dominio Amazon Elasticsearch Service

Il `create-elasticsearch-domain` comando seguente crea un nuovo dominio Amazon Elasticsearch Service all'interno di un VPC e limita l'accesso a un singolo utente. Amazon ES deduce l'VPCID dalla sottorete e dal gruppo di sicurezza specificati. 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
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Creating and Managing Amazon Elasticsearch Service Domains nella Amazon Elasticsearch Service Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [CreateElasticsearchDomain](#) AWS CLI

## describe-elasticsearch-domain-config

Il seguente esempio di codice mostra come utilizzare `describe-elasticsearch-domain-config`.

### AWS CLI

Per ottenere i dettagli sulla configurazione del dominio

L'esempio seguente fornisce i dettagli di configurazione per un determinato dominio, insieme alle informazioni sullo stato per ogni singolo componente del dominio.

```

aws es describe-elasticsearch-domain-config \
  --domain-name cli-example

```

## Output:

```
{
  "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
  }
}
}
```

Per ulteriori informazioni, consulta [Creating and Managing Amazon Elasticsearch Service Domains nella Amazon Elasticsearch Service Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [DescribeElasticsearchDomainConfig](#) AWS CLI

## describe-elasticsearch-domain

Il seguente esempio di codice mostra come utilizzare `describe-elasticsearch-domain`.

### AWS CLI

Per ottenere dettagli per un singolo dominio

L'esempio seguente fornisce i dettagli di configurazione per un determinato dominio.

```
aws es describe-elasticsearch-domain \
  --domain-name cli-example
```

Output:

```
{
  "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": [{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Principal": {"AWS": "*"}, "Action": "es:*", "Resource": "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,
      "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
    }
  }
}

```

Per ulteriori informazioni, consulta [Creating and Managing Amazon Elasticsearch Service Domains nella Amazon Elasticsearch Service Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [DescribeElasticsearchDomain](#) AWS CLI

## describe-elasticsearch-domains

Il seguente esempio di codice mostra come utilizzare `describe-elasticsearch-domains`.

### AWS CLI

Per ottenere dettagli su uno o più domini

L'`describe-elasticsearch-domains` seguente fornisce i dettagli di configurazione per uno o più domini.

```
aws es describe-elasticsearch-domains \  
  --domain-names cli-example-1 cli-example-2
```

Output:

```
{  
  "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
  }
}
]
}

```

Per ulteriori informazioni, consulta [Creating and Managing Amazon Elasticsearch Service Domains nella Amazon Elasticsearch Service Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [DescribeElasticsearchDomains](#) AWS CLI

## describe-reserved-elasticsearch-instances

Il seguente esempio di codice mostra come utilizzare `describe-reserved-elasticsearch-instances`.

### AWS CLI

Per visualizzare tutte le istanze riservate

L'`describe-elasticsearch-domains` esempio seguente fornisce un riepilogo di tutte le istanze prenotate in una regione.

```
aws es describe-reserved-elasticsearch-instances
```

Output:

```

{
  "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"
}]
}
```

Per ulteriori informazioni, consulta [Reserved Instances](#) nella Amazon Elasticsearch Service Developer Guide.

- Per API i dettagli, consulta Command [DescribeReservedElasticsearchInstances](#) Reference AWS CLI .

## list-domain-names

Il seguente esempio di codice mostra come utilizzare `list-domain-names`.

### AWS CLI

Per elencare tutti i domini

L'`list-domain-names` esempio seguente fornisce un breve riepilogo di tutti i domini della regione.

```
aws es list-domain-names
```

Output:

```
{
  "DomainNames": [{
    "DomainName": "cli-example-1"
  }],
}
```

```
    {  
      "DomainName": "cli-example-2"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Creating and Managing Amazon Elasticsearch Service Domains nella Amazon Elasticsearch Service Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [ListDomainNames](#)AWS CLI

## AWS OpsWorks esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS OpsWorks.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

### Azioni

#### **assign-instance**

Il seguente esempio di codice mostra come utilizzare `assign-instance`.

AWS CLI

Per assegnare un'istanza registrata a un livello

L'esempio seguente assegna un'istanza registrata a un livello personalizzato.

```
aws opsworks --region us-east-1 assign-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2 --layer-ids 26cf1d32-6876-42fa-bbf1-9cad0bfff938
```

Output: Nessuno.

Ulteriori informazioni

Per ulteriori informazioni, consulta Assegnazione di un'istanza registrata a un livello nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [AssignInstance](#) in AWS CLI Command Reference.

## **assign-volume**

Il seguente esempio di codice mostra come utilizzare `assign-volume`.

AWS CLI

Per assegnare un volume registrato a un'istanza

L'esempio seguente assegna un volume Amazon Elastic Block Store (AmazonEBS) registrato a un'istanza. Il volume è identificato dall'ID del volume, che è GUID quello assegnato quando si AWS OpsWorks registra il volume con uno stack, non dall'ID del volume Amazon Elastic Compute Cloud (AmazonEC2). Prima di eseguire `assign-volume`, devi prima eseguire l'operazione `update-volume` per assegnare un punto di montaggio al volume.

```
aws opsworks --region us-east-1 assign-volume --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2 --volume-id 26cf1d32-6876-42fa-bbf1-9cad0bfff938
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta Assigning Amazon EBS Volumes to an Instance nella AWS OpsWorks User Guide.

- Per API i dettagli, consulta [AssignVolume AWS CLI](#) Command Reference.

## **associate-elastic-ip**

Il seguente esempio di codice mostra come utilizzare `associate-elastic-ip`.

AWS CLI

Per associare un indirizzo IP elastico a un'istanza

L'esempio seguente associa un indirizzo IP elastico a un'istanza specificata.

```
aws opsworks --region us-east-1 associate-elastic-ip --instance-id dfe18b02-5327-493d-91a4-c5c0c448927f --elastic-ip 54.148.130.96
```

Output: Nessuno.

Ulteriori informazioni

Per ulteriori informazioni, vedere Resource Management nella Guida AWS OpsWorks per l'utente.

- Per API i dettagli, vedere [AssociateElasticIp](#) in AWS CLI Command Reference.

## **attach-elastic-load-balancer**

Il seguente esempio di codice mostra come utilizzare `attach-elastic-load-balancer`.

AWS CLI

Per collegare un sistema di bilanciamento del carico a un livello

L'esempio seguente collega un sistema di bilanciamento del carico, identificato dal relativo nome, a un livello specificato.

```
aws opsworks --region us-east-1 attach-elastic-load-balancer --elastic-load-balancer-name Java-LB --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta Elastic Load Balancing nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, consulta [AttachElasticLoadBalancer AWS CLI](#) Command Reference.

## **create-app**

Il seguente esempio di codice mostra come utilizzare `create-app`.

AWS CLI

Esempio 1: creare un'app

L'esempio seguente crea un'PHPApp denominata SimplePHPApp dal codice memorizzato in un GitHub repository. Il comando utilizza la forma abbreviata della definizione del codice sorgente dell'applicazione.

```
aws opsworks create-app \  
  --region us-east-1 \  
  --stack-id f6673d70-32e6-4425-8999-265dd002fec7 \  
  --name SimplePHPApp \  
  --type php \  
  --app-source Type=git,Url=git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git,Revision=version1
```

Output:

```
{  
  "AppId": "6cf5163c-a951-444f-a8f7-3716be75f2a2"  
}
```

Esempio 2: creare un'app con un database allegato

L'esempio seguente crea un'JSPApp dal codice memorizzato nell'archivio.zip in un bucket S3 pubblico. Collega un'istanza RDS DB che funge da archivio dati dell'app. Le fonti dell'applicazione e del database sono definite in JSON file separati che si trovano nella directory da cui si esegue il comando.

```
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
```

Le informazioni sulla fonte dell'applicazione sono in `appsource.json` e contengono quanto segue.

```
{  
  "Type": "archive",  
  "Url": "https://s3.amazonaws.com/opsworks-demo-assets/simplejsp.zip"  
}
```

Le informazioni sull'origine del database sono `datasource.json` incluse e contengono quanto segue.

```
[
  {
    "Type": "RdsDbInstance",
    "Arn": "arn:aws:rds:us-west-2:123456789012:db:clitestdb",
    "DatabaseName": "mydb"
  }
]
```

Nota: per un'istanza RDS DB, è necessario innanzitutto `register-rds-db-instance` registrare l'istanza nello stack. Per le istanze di My SQL App Server, imposta su `Type: OpsworksMySQLInstance`. Queste istanze sono create da AWS OpsWorks, quindi non devono essere registrate.

Output:

```
{
  "AppId": "26a61ead-d201-47e3-b55c-2a7c666942f8"
}
```

Per ulteriori informazioni, consulta [Aggiungere app](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [CreateApp](#) in AWS CLI Command Reference.

## create-deployment

Il seguente esempio di codice mostra come utilizzare `create-deployment`.

### AWS CLI

Esempio 1: per distribuire app ed eseguire comandi stack

Gli esempi seguenti mostrano come utilizzare il `create-deployment` comando per distribuire app ed eseguire comandi stack. Notate che i caratteri di virgoletta (") nell'JSON oggetto che specifica il comando sono tutti preceduti da caratteri di escape (\). Senza i caratteri di escape, il comando potrebbe restituire un errore non valido. JSON

L'`create-deployment` esempio seguente distribuisce un'app in uno stack specificato.

```
aws opsworks create-deployment \  
  --stack-id cfb7e082-ad1d-4599-8e81-de1c39ab45bf \  
  --app-id 307be5c8-d55d-47b5-bd6e-7bd417c6c7eb \  
  --command "{\"Name\": \"deploy\"}"
```

Output:

```
{  
  "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"  
}
```

Esempio 2: distribuire un'app Rails e migrare il database

Il create-deployment comando seguente distribuisce un'app Ruby on Rails in uno stack specificato e migra il database.

```
aws opsworks create-deployment \  
  --stack-id cfb7e082-ad1d-4599-8e81-de1c39ab45bf \  
  --app-id 307be5c8-d55d-47b5-bd6e-7bd417c6c7eb \  
  --command "{\"Name\": \"deploy\", \"Args\": {\"migrate\": [\"true\"]}}"
```

Output:

```
{  
  "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"  
}
```

Per ulteriori informazioni sulla distribuzione, consulta [Deploying](#) Apps nella Guida per l'utente.AWS OpsWorks

Esempio 3: Esegui una ricetta

Il create-deployment comando seguente esegue una ricetta personalizzataphpapp: :appsetup, sulle istanze in uno stack specificato.

```
aws opsworks create-deployment \  
  --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb \  
  --command "{\"Name\": \"execute_recipes\", \"Args\": {\"recipes\": [\"phpapp::appsetup\"]}}"
```

Output:



```
{
  "DeploymentId": "5cbaa7b9-4e09-4e53-aa1b-314fbd106038"
}
```

Per ulteriori informazioni, consulta [Run Stack Commands nella Guida](#) per l'AWS OpsWorks utente.

#### Esempio 4: installazione delle dipendenze

Il `create-deployment` comando seguente installa le dipendenze, come pacchetti o gemme Ruby, sulle istanze in uno stack specificato.

```
aws opsworks create-deployment \
  --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb \
  --command "{\"Name\":\"install_dependencies\"}"
```

Output:

```
{
  "DeploymentId": "aef5b255-8604-4928-81b3-9b0187f962ff"
}
```

Per ulteriori informazioni, consulta [Run Stack](#) Commands nella Guida per l'utente AWS OpsWorks

- Per API i dettagli, vedere [CreateDeployment](#) in AWS CLI Command Reference.

## create-instance

Il seguente esempio di codice mostra come utilizzare `create-instance`.

### AWS CLI

Per creare un'istanza

Il `create-instance` comando seguente crea un'istanza Amazon Linux `m1.large` denominata `myinstance1` in uno stack specificato. L'istanza viene assegnata a un livello.

```
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"
```

Per utilizzare un nome generato automaticamente, chiamate `get-hostname-suggestion`, che genera un nome host basato sul tema specificato al momento della creazione dello stack. Passate quindi quel nome all'argomento `hostname`.

Output:

```
{
  "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

Ulteriori informazioni

Per ulteriori informazioni, consulta [Aggiungere un'istanza a un livello nella Guida per l'AWS OpsWorks utente](#).

- Per API i dettagli, vedere [CreateInstance](#) in AWS CLI Command Reference.

## create-layer

Il seguente esempio di codice mostra come utilizzare `create-layer`.

AWS CLI

Per creare un livello

Il `create-layer` comando seguente crea un livello di PHP App Server denominato `MyPHPLayer` in uno stack specificato.

```
aws opsworks create-layer --region us-east-1 --stack-  
id f6673d70-32e6-4425-8999-265dd002fec7 --type php-app --name MyPHPLayer --  
shortname myphpLayer
```

Output:

```
{
  "LayerId": "0b212672-6b4b-40e4-8a34-5a943cf2e07a"
}
```

Ulteriori informazioni

Per ulteriori informazioni, consulta [Come creare un livello nella Guida per l'AWS OpsWorks utente](#).

- Per API i dettagli, vedere [CreateLayer](#) in AWS CLI Command Reference.

## create-server

Il seguente esempio di codice mostra come utilizzare `create-server`.

### AWS CLI

Per creare un server

L'esempio seguente crea un nuovo server Chef Automate denominato `automate-06` nella regione predefinita. Tieni presente che i valori predefiniti vengono utilizzati per la maggior parte delle altre impostazioni, come il numero di backup da conservare e gli orari di avvio della manutenzione e del backup. Prima di eseguire un `create-server` comando, completa i prerequisiti in [Getting Started with AWS OpsWorks for Chef Automate nella Guida per l'utente di AWS Opsworks for Chef Automate](#).

```
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"
```

Output:

```
{
  "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" ]
  }
}

```

Per ulteriori informazioni, consulta la sezione di riferimento di [CreateServer](#) Chef Automate AWS OpsWorks . API

- Per API i dettagli, vedere [CreateServer](#) in AWS CLI Command Reference.

## create-stack

Il seguente esempio di codice mostra come utilizzare `create-stack`.

### AWS CLI

Per creare una pila

Il `create-stack` comando seguente crea uno stack denominato 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

```

I parametri `service-role-arn` e `default-instance-profile-arn` sono obbligatori. Di solito usi quelli AWS OpsWorks creati per te quando crei il tuo primo stack. Per ottenere Amazon Resource Names (ARNs) per il tuo account, vai alla IAM console, scegli Roles nel pannello di navigazione, scegli il ruolo o il profilo e scegli la Summary scheda.

Output:

```
{
  "StackId": "f6673d70-32e6-4425-8999-265dd002fec7"
}
```

### Ulteriori informazioni

Per ulteriori informazioni, consulta [Creare un nuovo stack](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [CreateStack](#) in AWS CLI Command Reference.

## create-user-profile

Il seguente esempio di codice mostra come utilizzare `create-user-profile`.

### AWS CLI

Per creare un profilo utente

Si importa un utente di AWS Identity and Access Manager (IAM) AWS OpsWorks chiamando `create-user-profile` per creare un profilo utente. L'esempio seguente crea un profilo utente per l' `cli-user-test` utente, che viene identificato da Amazon Resource Name (ARN). L'esempio assegna all'utente un SSH nome utente di `myusername` e abilita l'autogestione, che consente all'utente di specificare una chiave SSH pubblica.

```
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
```

Output:

```
{
  "IamUserArn": "arn:aws:iam::123456789102:user/cli-user-test"
}
```

Suggerimento: Questo comando importa un IAM utente in AWS OpsWorks, ma solo con le autorizzazioni concesse dalle politiche allegate. È possibile concedere autorizzazioni per stack utilizzando il AWS OpsWorks comando. `set-permissions`

### Ulteriori informazioni

Per ulteriori informazioni, vedere Importazione di utenti AWS OpsWorks nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [CreateUserProfile](#) in AWS CLI Command Reference.

## delete-app

Il seguente esempio di codice mostra come utilizzare `delete-app`.

### AWS CLI

Per eliminare un'app

L'esempio seguente elimina un'app specificata, identificata dal relativo ID dell'app. È possibile ottenere un ID dell'app accedendo alla pagina dei dettagli dell'app sulla AWS OpsWorks console o eseguendo il `describe-apps` comando.

```
aws opsworks delete-app --region us-east-1 --app-id 577943b9-2ec1-4baf-a7bf-1d347601edc5
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta App nella Guida AWS OpsWorks per l'utente.

- Per API i dettagli, vedere [DeleteApp](#) in AWS CLI Command Reference.

## delete-instance

Il seguente esempio di codice mostra come utilizzare `delete-instance`.

### AWS CLI

Per eliminare un'istanza

L'`delete-instance` esempio seguente elimina un'istanza specificata, identificata dal relativo ID di istanza. È possibile trovare un ID di istanza aprendo la pagina dei dettagli dell'istanza nella AWS OpsWorks console o eseguendo il `describe-instances` comando.

Se l'istanza è online, devi prima interromperla chiamando `stop-instance`, quindi devi attendere che l'istanza si fermi. Esegui `describe-instances` per controllare lo stato dell'istanza.

Per rimuovere i EBS volumi Amazon o gli indirizzi IP elastici dell'istanza, aggiungi rispettivamente `--delete-elastic-ip` gli argomenti `--delete-volumes` o.

```
aws opsworks delete-instance \  
  --region us-east-1 \  
  --instance-id 3a21cfac-4a1f-4ce2-a921-b2cfba6f7771
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione AWS OpsWorks delle istanze nella Guida](#) per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DeleteInstance](#) in AWS CLI Command Reference.

## delete-layer

Il seguente esempio di codice mostra come utilizzare `delete-layer`.

### AWS CLI

Per eliminare un livello

L'esempio seguente elimina un livello specificato, identificato dal relativo ID di livello. È possibile ottenere un ID di livello accedendo alla pagina dei dettagli del livello sulla AWS OpsWorks console o eseguendo il `describe-layers` comando.

Nota: prima di eliminare un livello, è necessario utilizzare `delete-instance` per eliminare tutte le istanze del livello.

```
aws opsworks delete-layer --region us-east-1 --layer-id a919454e-b816-4598-  
b29a-5796afb498ed
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta [Eliminazione AWS OpsWorks delle istanze nella Guida](#) per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DeleteLayer](#) in AWS CLI Command Reference.

## delete-stack

Il seguente esempio di codice mostra come utilizzare `delete-stack`.

### AWS CLI

Per eliminare una pila

L'esempio seguente elimina uno stack specificato, identificato dal relativo ID dello stack. È possibile ottenere un ID dello stack facendo clic su Impostazioni dello stack sulla AWS OpsWorks console o eseguendo il comando `describe-stacks`

Nota: prima di eliminare un layer, è necessario utilizzare `delete-layer` per eliminare tutte le app `delete-app` `delete-instance`, le istanze e i layer dello stack.

```
aws opsworks delete-stack --region us-east-1 --stack-id 154a9d89-7e9e-433b-8de8-617e53756c84
```

Output: Nessuno.

Ulteriori informazioni

Per ulteriori informazioni, consulta Shut Down a Stack nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DeleteStack](#) in AWS CLI Command Reference.

## delete-user-profile

Il seguente esempio di codice mostra come utilizzare `delete-user-profile`.

### AWS CLI

Per eliminare un profilo utente e rimuovere un IAM utente da AWS OpsWorks

L'esempio seguente elimina il profilo utente per un utente specificato di AWS Identity and Access Management (IAM), identificato da Amazon Resource Name (ARN). L'operazione rimuove l'utente da AWS OpsWorks, ma non lo IAM elimina. È necessario utilizzare la IAM CLI console o API per tale attività.

```
aws opsworks --region us-east-1 delete-user-profile --iam-user-arn arn:aws:iam::123456789102:user/cli-user-test
```



Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, vedere Importazione di utenti AWS OpsWorks nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DeleteUserProfile](#) in AWS CLI Command Reference.

## **deregister-elastic-ip**

Il seguente esempio di codice mostra come utilizzare `deregister-elastic-ip`.

AWS CLI

Per annullare la registrazione di un indirizzo IP elastico da uno stack

L'esempio seguente annulla la registrazione di un indirizzo IP elastico, identificato dal relativo indirizzo IP, dal relativo stack.

```
aws opsworks deregister-elastic-ip --region us-east-1 --elastic-ip 54.148.130.96
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta Annullamento della registrazione degli indirizzi IP elastici nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DeregisterElasticIp](#) in AWS CLI Command Reference.

## **deregister-instance**

Il seguente esempio di codice mostra come utilizzare `deregister-instance`.

AWS CLI

Per annullare la registrazione di un'istanza registrata da uno stack

Il `deregister-instance` comando seguente annulla la registrazione di un'istanza registrata dal relativo stack.

```
aws opsworks --region us-east-1 deregister-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta Annullamento della registrazione di un'istanza registrata nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DeregisterInstance](#) in AWS CLI Command Reference.

## deregister-rds-db-instance

Il seguente esempio di codice mostra come utilizzare `deregister-rds-db-instance`.

AWS CLI

Per annullare la registrazione di un'istanza Amazon RDS DB da uno stack

L'esempio seguente annulla la registrazione di un'istanza RDS DB, identificata da `its`, dal relativo stack. ARN

```
aws opsworks deregister-rds-db-instance --region us-east-1 --rds-db-instance-arn arn:aws:rds:us-west-2:123456789012:db:clitestdb
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta Annullamento della registrazione di Amazon RDS Instances nella Guida per l'ASW OpsWorks utente.

ID dell'istanza: `clitestdb` Nome utente principale: `clitestdb` Master: `some23!` PWD `pwd` Nome DB: `mydb`  
aws opsworks deregister-rds-db-instance --region `us-east-1` -- arn:aws:rds:us-west-2:645732743964:db:clitestdb rds-db-instance-arn

- Per i API dettagli [DeregisterRdsDbInstance](#), vedere in AWS CLI Command Reference.

## deregister-volume

Il seguente esempio di codice mostra come utilizzare `deregister-volume`.

## AWS CLI

Per annullare la registrazione di un volume Amazon EBS

L'esempio seguente annulla la registrazione di un EBS volume dal relativo stack. Il volume è identificato dal relativo ID di volume, che è GUID quello AWS OpsWorks assegnato al momento della registrazione del volume nello stack, non dall'ID del volume. EC2

```
aws opsworks deregister-volume --region us-east-1 --volume-id 5c48ef52-3144-4bf5-beaa-fda4deb23d4d
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta Annullamento della registrazione di Amazon EBS Volumes nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, consulta AWS CLI Command [DeregisterVolume](#)Reference.

## describe-apps

Il seguente esempio di codice mostra come utilizzare `describe-apps`.

### AWS CLI

Per descrivere le app

Il `describe-apps` comando seguente descrive le app in uno stack specificato.

```
aws opsworks describe-apps \  
  --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a \  
  --region us-east-1
```

Output:

```
{  
  "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"
}
]
```

Per ulteriori informazioni, consulta App nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeApps](#) in AWS CLI Command Reference.

## describe-commands

Il seguente esempio di codice mostra come utilizzare `describe-commands`.

### AWS CLI

Per descrivere i comandi

Il `describe-commands` comando seguente descrive i comandi in un'istanza specificata.

```
aws opsworks describe-commands \  
  --instance-id 8c2673b9-3fe5-420d-9cfa-78d875ee7687 \  
  --region us-east-1
```

Output:

```
{  
  "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
  }
]
}

```

Per ulteriori informazioni, consulta AWS OpsWorks Lifecycle Events nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeCommands](#) in AWS CLI Command Reference.

## describe-deployments

Il seguente esempio di codice mostra come utilizzare `describe-deployments`.

### AWS CLI

Per descrivere le distribuzioni

Il `describe-deployments` comando seguente descrive le distribuzioni in uno stack specificato.

```
aws opsworks --region us-east-1 describe-deployments --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a
```

Output:

```
{
  "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"
    }
  ]
}
```

```
}
```

### Ulteriori informazioni

Per ulteriori informazioni, consulta [Deploying Apps](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeDeployments](#) in AWS CLI Command Reference.

## describe-elastic-ips

Il seguente esempio di codice mostra come utilizzare `describe-elastic-ips`.

### AWS CLI

Per descrivere le istanze Elastic IP

Il `describe-elastic-ips` comando seguente descrive gli indirizzi IP elastici in un'istanza specificata.

```
aws opsworks --region us-east-1 describe-elastic-ips --instance-id b62f3e04-  
e9eb-436c-a91f-d9e9a396b7b0
```

### Output:

```
{  
  "ElasticIps": [  
    {  
      "Ip": "192.0.2.0",  
      "Domain": "standard",  
      "Region": "us-west-2"  
    }  
  ]  
}
```

### Ulteriori informazioni

Per ulteriori informazioni, consulta [Istanze](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeElasticIps](#) in AWS CLI Command Reference.

## describe-elastic-load-balancers

Il seguente esempio di codice mostra come utilizzare `describe-elastic-load-balancers`.

## AWS CLI

Per descrivere i bilanciatori di carico elastici di uno stack

Il `describe-elastic-load-balancers` comando seguente descrive i bilanciatori di carico di uno stack specificato.

```
aws opsworks --region us-west-2 describe-elastic-load-balancers --stack-id 6f4660e5-37a6-4e42-bfa0-1358ebd9c182
```

Output: questo particolare stack ha un sistema di bilanciamento del carico.

```
{
  "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"
    }
  ]
}
```

### Ulteriori informazioni

Per ulteriori informazioni, consulta App nella Guida AWS OpsWorks per l'utente.

- Per API i dettagli, vedere [DescribeElasticLoadBalancers](#) in AWS CLI Command Reference.

## describe-instances

Il seguente esempio di codice mostra come utilizzare `describe-instances`.



## AWS CLI

Per descrivere le istanze

Il `describe-instances` comando seguente descrive le istanze in uno stack specificato:

```
aws opsworks --region us-east-1 describe-instances --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Output: il seguente esempio di output riguarda uno stack con due istanze. La prima è un'EC2istanza registrata e la seconda è stata creata da AWS OpsWorks

```
{
  "Instances": [
    {
      "StackId": "71c7ca72-55ae-4b6a-8ee1-a8dcdded3fa0f",
      "PrivateDns": "ip-10-31-39-66.us-west-2.compute.internal",
      "LayerIds": [
        "26cf1d32-6876-42fa-bbf1-9cadc0bff938"
      ],
      "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"  
  }  
]  
}
```

## Ulteriori informazioni

Per ulteriori informazioni, consulta *Istanze* nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeInstances](#) in AWS CLI Command Reference.

## describe-layers

Il seguente esempio di codice mostra come utilizzare `describe-layers`.

### AWS CLI

Per descrivere i livelli di una pila

Il `describe-layers` comando seguente descrive i livelli di uno stack specificato:

```
aws opsworks --region us-east-1 describe-layers --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a
```

### Output:

```
{  
  "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": []
    }
]
}

```

## Ulteriori informazioni

Per ulteriori informazioni, consulta Layers nella Guida AWS OpsWorks per l'utente.

- Per API i dettagli, vedere [DescribeLayers](#) in AWS CLI Command Reference.

## describe-load-based-auto-scaling

Il seguente esempio di codice mostra come utilizzare `describe-load-based-auto-scaling`.

### AWS CLI

Per descrivere la configurazione di ridimensionamento basata sul carico di un livello

L'esempio seguente descrive la configurazione di ridimensionamento basato sul carico di un livello specificato. Il layer è identificato dal relativo ID di livello, che puoi trovare nella pagina dei dettagli del layer o eseguendo `describe-layers`

```
aws opsworks describe-load-based-auto-scaling --region us-east-1 --layer-ids 6bec29c9-c866-41a0-aba5-fa3e374ce2a1
```

Output: il layer di esempio ha una singola istanza basata sul caricamento.

```

{
  "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"  
}  
]  
}
```

## Ulteriori informazioni

Per ulteriori informazioni, consulta [How Automatic Load-based Scaling](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeLoadBasedAutoScaling](#) in AWS CLI Command Reference.

## describe-my-user-profile

Il seguente esempio di codice mostra come utilizzare `describe-my-user-profile`.

### AWS CLI

Per ottenere il profilo di un utente

L'esempio seguente mostra come ottenere il profilo dell'utente AWS Identity and Access Management (IAM) che esegue il comando.

```
aws opsworks --region us-east-1 describe-my-user-profile
```

Risultato: per brevità, la maggior parte della chiave SSH pubblica dell'utente viene sostituita da puntini di sospensione (...).

```
{  
  "UserProfile": {  
    "IamUserArn": "arn:aws:iam::123456789012:user/myusername",  
    "SshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAABJQ...3LQ4aX9jpxQw== rsa-  
key-20141104",  
    "Name": "myusername",  
    "SshUsername": "myusername"  
  }  
}
```



## Ulteriori informazioni

Per ulteriori informazioni, vedere Importazione di utenti AWS OpsWorks nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeMyUserProfile](#) in AWS CLI Command Reference.

## describe-permissions

Il seguente esempio di codice mostra come utilizzare `describe-permissions`.

### AWS CLI

Per ottenere il livello di autorizzazione per stack di AWS OpsWorks un utente

L'esempio seguente mostra come ottenere il livello di autorizzazione di un utente di AWS Identity and Access Management (IAM) su uno stack specificato.

```
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
```

### Output:

```
{
  "Permissions": [
    {
      "StackId": "d72553d4-8727-448c-9b00-f024f0ba1b06",
      "IamUserArn": "arn:aws:iam::123456789012:user/cli-user-test",
      "Level": "manage",
      "AllowSudo": true,
      "AllowSsh": true
    }
  ]
}
```

## Ulteriori informazioni

Per ulteriori informazioni, consulta Concessione dei livelli di autorizzazione per stack nella Guida per l'utente AWS OpsWorks

- Per i API dettagli, vedere in Command Reference. [DescribePermissions](#) AWS CLI

## describe-raid-arrays

Il seguente esempio di codice mostra come utilizzare `describe-raid-arrays`.

### AWS CLI

Per descrivere gli RAID array

L'esempio seguente descrive gli RAID array collegati alle istanze in uno stack specificato.

```
aws opsworks --region us-east-1 describe-raid-arrays --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06
```

Output: quello che segue è l'output di uno stack con un array RAID

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta EBS Volumes nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeRaidArrays](#) in AWS CLI Command Reference.

## describe-rds-db-instances

Il seguente esempio di codice mostra come utilizzare `describe-rds-db-instances`.

## AWS CLI

Per descrivere le istanze Amazon RDS registrate di uno stack

L'esempio seguente descrive le RDS istanze Amazon registrate con uno stack specificato.

```
aws opsworks --region us-east-1 describe-rds-db-instances --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06
```

Output: quello che segue è l'output di uno stack con un'istanza registrata. RDS

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione Gestione delle risorse nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeRdsDbInstances](#) in AWS CLI Command Reference.

## describe-stack-provisioning-parameters

Il seguente esempio di codice mostra come utilizzare `describe-stack-provisioning-parameters`.

### AWS CLI

Per restituire i parametri di provisioning per uno stack

L'`describe-stack-provisioning-parameters` esempio seguente restituisce i parametri di provisioning per uno stack specificato. I parametri di provisioning includono impostazioni come la

posizione di installazione dell'agente e la chiave pubblica OpsWorks utilizzata per gestire l'agente sulle istanze di uno stack.

```
aws opsworks describe-stack-provisioning-parameters \  
  --stack-id 62744d97-6faf-4ecb-969b-a086fEXAMPLE
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Run Stack Commands](#) nella Guida per l'utente AWS OpsWorks

- Per API i dettagli, vedere [DescribeStackProvisioningParameters](#) in AWS CLI Command Reference.

## describe-stack-summary

Il seguente esempio di codice mostra come utilizzare `describe-stack-summary`.

### AWS CLI

Per descrivere la configurazione di uno stack

Il `describe-stack-summary` comando seguente restituisce un riepilogo della configurazione dello stack specificato.

```
aws opsworks --region us-east-1 describe-stack-summary --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Output:

```
{
  "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/"
  }
}
```

### Ulteriori informazioni

Per ulteriori informazioni, consulta [Stacks](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeStackSummary](#) in AWS CLI Command Reference.

## describe-stacks

Il seguente esempio di codice mostra come utilizzare `describe-stacks`.

### AWS CLI

Per descrivere le pile

Il `describe-stacks` comando seguente descrive gli stack di un account.

```
aws opsworks --region us-east-1 describe-stacks
```

Output:

```
{
  "Stacks": [
    {
      "ServiceRoleArn": "arn:aws:iam::444455556666:role/aws-opsworks-service-role",
      "StackId": "aeb7523e-7c8b-49d4-b866-03aae9d4fbcf",
      "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"
}
]
}

```

## Ulteriori informazioni

Per ulteriori informazioni, consulta [Stacks](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeStacks](#) in AWS CLI Command Reference.

## describe-timebased-auto-scaling

Il seguente esempio di codice mostra come utilizzare `describe-timebased-auto-scaling`.

### AWS CLI

Per descrivere la configurazione di scalabilità basata sul tempo di un'istanza

L'esempio seguente descrive la configurazione di scalabilità basata sul tempo di un'istanza specificata. L'istanza è identificata dal relativo ID, che puoi trovare nella pagina dei dettagli delle istanze o eseguendo `describe-instances`

```
aws opsworks describe-time-based-auto-scaling --region us-east-1 --instance-ids 701f2ffe-5d8e-4187-b140-77b75f55de8d
```

Output: l'esempio ha una singola istanza basata sul tempo.

```
{
```

```
"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"  
      }  
    }  
  }  
]
```

### Ulteriori informazioni

Per ulteriori informazioni, consulta [Come funziona il ridimensionamento automatico basato sul tempo](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeTimebasedAutoScaling](#) in AWS CLI Command Reference.

## describe-user-profiles

Il seguente esempio di codice mostra come utilizzare `describe-user-profiles`.

### AWS CLI

Per descrivere i profili utente

Il `describe-user-profiles` comando seguente descrive i profili utente dell'account.

```
aws opsworks --region us-east-1 describe-user-profiles
```

Output:

```
{
```



```
"UserProfiles": [  
  {  
    "IamUserArn": "arn:aws:iam::123456789012:user/someuser",  
    "SshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEak0uP7i80q3Cko...",  
    "AllowSelfManagement": true,  
    "Name": "someuser",  
    "SshUsername": "someuser"  
  },  
  {  
    "IamUserArn": "arn:aws:iam::123456789012:user/cli-user-test",  
    "AllowSelfManagement": true,  
    "Name": "cli-user-test",  
    "SshUsername": "myusername"  
  }  
]  
}
```

## Ulteriori informazioni

Per ulteriori informazioni, consulta la sezione Gestione AWS OpsWorks degli utenti nella Guida AWS OpsWorks per l'utente.

- Per API i dettagli, vedere [DescribeUserProfiles](#) in AWS CLI Command Reference.

## describe-volumes

Il seguente esempio di codice mostra come utilizzare `describe-volumes`.

### AWS CLI

Per descrivere i volumi di una pila

L'esempio seguente descrive i volumi di uno stack. EBS

```
aws opsworks --region us-east-1 describe-volumes --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Output:

```
{  
  "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
}
]
}
```

### Ulteriori informazioni

Per ulteriori informazioni, consulta la sezione Gestione delle risorse nella Guida AWS OpsWorks per l'utente.

- Per API i dettagli, vedere [DescribeVolumes](#) in AWS CLI Command Reference.

## detach-elastic-load-balancer

Il seguente esempio di codice mostra come utilizzare `detach-elastic-load-balancer`.

### AWS CLI

Per scollegare un sistema di bilanciamento del carico dal relativo livello

L'esempio seguente rimuove un sistema di bilanciamento del carico, identificato dal relativo nome, dal relativo livello.

```
aws opsworks --region us-east-1 detach-elastic-load-balancer --elastic-load-balancer-name Java-LB --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

Uscita: nessuna.

### Ulteriori informazioni

Per ulteriori informazioni, consulta Elastic Load Balancing nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, consulta [DetachElasticLoadBalancer](#) AWS CLI Command Reference.

## disassociate-elastic-ip

Il seguente esempio di codice mostra come utilizzare `disassociate-elastic-ip`.

### AWS CLI

Per dissociare un indirizzo IP elastico da un'istanza

L'esempio seguente dissocia un indirizzo IP elastico da un'istanza specificata.

```
aws opsworks --region us-east-1 disassociate-elastic-ip --elastic-ip 54.148.130.96
```

Output: Nessuno.

### Ulteriori informazioni

Per ulteriori informazioni, consulta la sezione Gestione delle risorse nella Guida AWS OpsWorks per l'utente.

- Per API i dettagli, vedere [DisassociateElasticIp](#) in AWS CLI Command Reference.

## get-hostname-suggestion

Il seguente esempio di codice mostra come utilizzare `get-hostname-suggestion`.

### AWS CLI

Per ottenere il nome host successivo per un livello

L'esempio seguente ottiene il nome host generato successivamente per un livello specificato. Il livello utilizzato per questo esempio è un livello Java Application Server con un'istanza. Il tema del nome host dello stack è quello predefinito, `Layer_Dependent`.

```
aws opsworks --region us-east-1 get-hostname-suggestion --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

Output:

```
{
  "Hostname": "java-app2",
  "LayerId": "888c5645-09a5-4d0e-95a8-812ef1db76a4"
}
```

## Ulteriori informazioni

Per ulteriori informazioni, consulta [Creare un nuovo stack](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [GetHostnameSuggestion](#) in AWS CLI Command Reference.

## reboot-instance

Il seguente esempio di codice mostra come utilizzare `reboot-instance`.

### AWS CLI

Per riavviare un'istanza

L'esempio seguente riavvia un'istanza.

```
aws opsworks --region us-east-1 reboot-instance --instance-id dfe18b02-5327-493d-91a4-c5c0c448927f
```

Uscita: nessuna.

## Ulteriori informazioni

Per ulteriori informazioni, consulta [Riavvio di un'istanza](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, consulta AWS CLI Command [RebootInstance](#) Reference.

## register-elastic-ip

Il seguente esempio di codice mostra come utilizzare `register-elastic-ip`.

### AWS CLI

Per registrare un indirizzo IP elastico con uno stack

L'esempio seguente registra un indirizzo IP elastico, identificato dal relativo indirizzo IP, con uno stack specificato.

Nota: l'indirizzo IP elastico deve trovarsi nella stessa area dello stack.

```
aws opsworks register-elastic-ip --region us-east-1 --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06 --elastic-ip 54.148.130.96
```

## Output

```
{
  "ElasticIp": "54.148.130.96"
}
```

## Ulteriori informazioni

Per ulteriori informazioni, consulta [Registrazione di indirizzi IP elastici con uno stack nella Guida per l'OpsWorks utente](#).

- Per API i dettagli, vedere [RegisterElasticIp](#) in AWS CLI Command Reference.

## **register-rds-db-instance**

Il seguente esempio di codice mostra come utilizzare `register-rds-db-instance`.

### AWS CLI

Per registrare un'RDSistanza Amazon con uno stack

L'esempio seguente registra un'istanza Amazon RDS DB, identificata dal relativo Amazon Resource Name (ARN), con uno stack specificato. Specificano inoltre il nome utente e la password principali dell'istanza. Nota che AWS OpsWorks non convalida nessuno di questi valori. Se uno dei due non è corretto, l'applicazione non sarà in grado di connettersi al database.

```
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
```

Uscita: nessuna.

## Ulteriori informazioni

Per ulteriori informazioni, consulta [Registrazione di RDS istanze Amazon con uno stack nella Guida per l'AWS OpsWorks utente](#).

- Per API i dettagli, consulta Command [RegisterRdsDbInstance](#) Reference AWS CLI .

## **register-volume**

Il seguente esempio di codice mostra come utilizzare `register-volume`.

## AWS CLI

Per registrare un EBS volume Amazon con uno stack

L'esempio seguente registra un EBS volume Amazon, identificato dal relativo ID di volume, con uno stack specificato.

```
aws opsworks register-volume --region us-east-1 --stack-id d72553d4-8727-448c-9b00-f024f0ba1b06 --ec-2-volume-id vol-295c1638
```

Output:

```
{  
  "VolumeId": "ee08039c-7cb7-469f-be10-40fb7f0c05e8"  
}
```

Ulteriori informazioni

Per ulteriori informazioni, consulta [Registrazione di EBS volumi Amazon con uno stack](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, consulta AWS CLI Command [RegisterVolume](#)Reference.

## register

Il seguente esempio di codice mostra come utilizzare `register`.

## AWS CLI

Per registrare istanze con uno stack

Gli esempi seguenti mostrano diversi modi per registrare istanze con uno stack creato all'esterno di Opsworks. AWS È possibile eseguire l'esecuzione `register` dall'istanza da registrare o da una workstation separata. Per ulteriori informazioni, consulta [Registrazione di istanze Amazon EC2 e locali](#) nella Guida per l'AWS OpsWorks utente.

Nota: per brevità, gli esempi omettono l'argomento. `region`

Per registrare un'EC2istanza Amazon

Per indicare che stai registrando un'EC2istanza, imposta l'`--infrastructure-class` argomento su. `ec2`

L'esempio seguente registra un'EC2istanza con lo stack specificato da una workstation separata. L'istanza è identificata dal relativo EC2 ID, `i-12345678`. L'esempio utilizza il SSH nome utente predefinito della workstation e tenta di accedere all'istanza utilizzando tecniche di autenticazione che non richiedono una password, come una SSH chiave privata predefinita. Se ciò non riesce, `register` richiede la password.

```
aws opsworks register --infrastructure-class=ec2 --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb i-12345678
```

L'esempio seguente registra un'EC2istanza con lo stack specificato da una workstation separata. Utilizza gli `--ssh-private-key` argomenti `--ssh-username` and per specificare in modo esplicito il SSH nome utente e il file di chiave privata utilizzati dal comando per accedere all'istanza. `ec2-user` è il nome utente standard per le istanze Amazon Linux. Utilizzalo `ubuntu` per le istanze di Ubuntu.

```
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
```

L'esempio seguente registra l'EC2istanza che esegue il `register` comando. Accedi all'istanza SSH ed esegui con l'`--local` argomento anziché `register` con l'ID dell'istanza o il nome host.

```
aws opsworks register --infrastructure-class ec2 --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --local
```

Per registrare un'istanza locale

Per indicare che stai registrando un'istanza locale, imposta l'`--infrastructure-class` argomento su `on-premises`

L'esempio seguente registra un'istanza locale esistente con uno stack specificato da una workstation separata. L'istanza è identificata dal relativo indirizzo IP, `192.0.2.3`. L'esempio utilizza il SSH nome utente predefinito della workstation e tenta di accedere all'istanza utilizzando tecniche di autenticazione che non richiedono una password, come una SSH chiave privata predefinita. Se ciò non riesce, `register` richiede la password.

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb 192.0.2.3
```

L'esempio seguente registra un'istanza locale con uno stack specificato da una workstation separata. L'istanza è identificata dal relativo nome host, `host1`. Gli `--override-...` argomenti vengono visualizzati direttamente AWS OpsWorks `webserver1` come nome host `192.0.2.3` e `10.0.0.2` come indirizzi IP pubblici e privati dell'istanza, rispettivamente.

```
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
```

L'esempio seguente registra un'istanza locale con uno stack specificato da una workstation separata. L'istanza è identificata dal relativo indirizzo IP. `register` accede all'istanza utilizzando il SSH nome utente e il file di chiave privata specificati.

```
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
```

L'esempio seguente registra un'istanza locale esistente con uno stack specificato da una workstation separata. Il comando accede all'istanza utilizzando una stringa di SSH comando personalizzata che specifica la SSH password e l'indirizzo IP dell'istanza.

```
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"
```

L'esempio seguente registra l'istanza locale che esegue il comando. `register` Accedi all'istanza SSH ed esegui con l'`--local` argomento anziché `register` con un ID di istanza o un nome host.

```
aws opsworks register --infrastructure-class on-premises --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb --local
```

Output: quello che segue è l'output tipico per la registrazione di un'EC2istanza.

```
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.
```

## Ulteriori informazioni

Per ulteriori informazioni, consulta [Registrazione di un'istanza con uno AWS OpsWorks stack](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, consulta [Register](#) in AWS CLI Command Reference.

## set-load-based-auto-scaling

Il seguente esempio di codice mostra come utilizzare `set-load-based-auto-scaling`.

### AWS CLI

Per impostare la configurazione di ridimensionamento basata sul carico per un livello

L'esempio seguente abilita il ridimensionamento basato sul carico per un livello specificato e imposta la configurazione per quel livello. È necessario utilizzare `create-instance` per aggiungere istanze basate sul caricamento al livello.

```
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
```

L'esempio inserisce le impostazioni della soglia di upscaling in un file separato nella directory di lavoro denominata `upscale.json`, che contiene quanto segue.

```
{  
  "InstanceCount": 2,  
  "ThresholdsWaitTime": 3,  
  "IgnoreMetricsTime": 3,  
  "CpuThreshold": 85,  
  "MemoryThreshold": 85,  
  "LoadThreshold": 85  
}
```

L'esempio inserisce le impostazioni della soglia di downscaling in un file separato nella directory di lavoro denominata `downscale.json`, che contiene quanto segue.

```
{  
  "InstanceCount": 2,  
  "ThresholdsWaitTime": 3,  
  "IgnoreMetricsTime": 3,  
  "CpuThreshold": 35,  
  "MemoryThreshold": 30,  
  "LoadThreshold": 30  
}
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, vedere [Using Automatic Load-based Scaling](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [SetLoadBasedAutoScaling](#) in AWS CLI Command Reference.

## set-permission

Il seguente esempio di codice mostra come utilizzare `set-permission`.

## AWS CLI

Per concedere livelli di autorizzazione per stack AWS OpsWorks

Quando si importa un utente AWS Identity and Access Management (IAM) AWS OpsWorks mediante chiamata `create-user-profile`, l'utente dispone solo delle autorizzazioni concesse dalle IAM politiche allegate. È possibile concedere AWS OpsWorks le autorizzazioni modificando le politiche di un utente. Tuttavia, spesso è più semplice importare un utente e quindi utilizzare il `set-permission` comando per concedere all'utente uno dei livelli di autorizzazione standard per ogni stack a cui l'utente dovrà accedere.

L'esempio seguente concede l'autorizzazione per lo stack specificato a un utente, identificato da Amazon Resource Name (ARN). L'esempio concede all'utente un livello di autorizzazione `Manage`, con `sudo` e `SSH` privilegi sulle istanze dello stack.

```
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
```

Output: Nessuno.

Ulteriori informazioni

Per ulteriori informazioni, consulta [Concessione delle autorizzazioni per stack AWS OpsWorks agli utenti](#) nella Guida per l'utente AWS OpsWorks

- Per i API dettagli, vedere in Command Reference. [SetPermission](#) AWS CLI

## set-time-based-auto-scaling

Il seguente esempio di codice mostra come utilizzare `set-time-based-auto-scaling`.

### AWS CLI

Per impostare la configurazione di ridimensionamento basata sul tempo per un livello

L'esempio seguente imposta la configurazione basata sul tempo per un'istanza specificata. È necessario innanzitutto utilizzare `create-instance` per aggiungere l'istanza al livello.

```
aws opsworks --region us-east-1 set-time-based-auto-scaling --instance-id 69b6237c-08c0-4edb-a6af-78f3d01cedf2 --auto-scaling-schedule file://schedule.json
```

L'esempio inserisce la pianificazione in un file separato nella directory di lavoro denominata `schedule.json`. Per questo esempio, l'istanza rimane attiva per alcune ore intorno a mezzogiorno UTC (Coordinated Universal Time) del lunedì e del martedì.

```
{
  "Monday": {
    "10": "on",
    "11": "on",
    "12": "on",
    "13": "on"
  },
  "Tuesday": {
    "10": "on",
    "11": "on",
    "12": "on",
    "13": "on"
  }
}
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, vedere [Utilizzo del ridimensionamento automatico basato sul tempo](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [SetTimeBasedAutoScaling](#) in AWS CLI Command Reference.

## start-instance

Il seguente esempio di codice mostra come utilizzare `start-instance`.

AWS CLI

Per avviare un'istanza

Il `start-instance` comando seguente avvia un'istanza specificata 24 ore su 24, 7 giorni su 7.

```
aws opsworks start-instance --instance-id f705ee48-9000-4890-8bd3-20eb05825aaf
```

Uscita: nessuna. Usa `describe-instances` per controllare lo stato dell'istanza.

Suggerimento Puoi avviare ogni istanza offline in uno stack con un solo comando chiamando `start-stack`.

Ulteriori informazioni

Per ulteriori informazioni, consulta [Avvio, arresto e riavvio manuale delle istanze 24 ore su 24, 7 giorni su 7](#) nella Guida per l'utente.AWS OpsWorks

- Per i API dettagli, consulta [Command Reference](#). [StartInstance](#)AWS CLI

## **start-stack**

Il seguente esempio di codice mostra come utilizzare `start-stack`.

AWS CLI

Per avviare le istanze di uno stack

L'esempio seguente avvia tutte le istanze di uno stack 24 ore su 24, 7 giorni su 7. Per avviare una particolare istanza, usa `start-instance`

```
aws opsworks --region us-east-1 start-stack --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta [Avvio di un'istanza](#) nella Guida AWS OpsWorks per l'utente.

- Per API i dettagli, consulta [StartStack AWS CLI](#)Command Reference.

## **stop-instance**

Il seguente esempio di codice mostra come utilizzare `stop-instance`.

AWS CLI

Per interrompere un'istanza

L'esempio seguente arresta un'istanza specificata, identificata dal relativo ID di istanza. È possibile ottenere un ID di istanza accedendo alla pagina dei dettagli dell'istanza sulla AWS OpsWorks console o eseguendo il `describe-instances` comando.

```
aws opsworks stop-instance --region us-east-1 --instance-id 3a21cfac-4a1f-4ce2-a921-b2cfba6f7771
```

È possibile riavviare un'istanza interrotta chiamando `start-instance` o eliminando l'istanza `delete-instance` chiamando.

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta [Arresto di un'istanza](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, consulta [StopInstance AWS CLI Command Reference](#).

## stop-stack

Il seguente esempio di codice mostra come utilizzare `stop-stack`.

AWS CLI

Per fermare le istanze di uno stack

L'esempio seguente interrompe tutte le istanze di uno stack 24 ore su 24, 7 giorni su 7. Per interrompere una particolare istanza, usa `stop-instance`

```
aws opsworks --region us-east-1 stop-stack --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

Uscita: nessuna uscita.

Ulteriori informazioni

Per ulteriori informazioni, consulta [Arresto di un'istanza](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, consulta [StopStack AWS CLI Command Reference](#).

## unassign-instance

Il seguente esempio di codice mostra come utilizzare `unassign-instance`.

AWS CLI

Per annullare l'assegnazione di un'istanza registrata dai suoi livelli

Il `unassign-instance` comando seguente annulla l'assegnazione di un'istanza dai livelli collegati.

```
aws opsworks --region us-east-1 unassign-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
```

Output: Nessuno.

Ulteriori informazioni

Per ulteriori informazioni, consulta Annullamento dell'assegnazione di un'istanza registrata nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [UnassignInstance](#) in AWS CLI Command Reference.

## **unassign-volume**

Il seguente esempio di codice mostra come utilizzare `unassign-volume`.

AWS CLI

Per annullare l'assegnazione di un volume alla sua istanza

L'esempio seguente annulla l'assegnazione di un volume Amazon Elastic Block Store EBS (Amazon) registrato alla sua istanza. Il volume è identificato dall'ID del volume, che è GUID quello assegnato quando si AWS OpsWorks registra il volume con uno stack, non dall'ID del volume Amazon Elastic Compute Cloud (AmazonEC2).

```
aws opsworks --region us-east-1 unassign-volume --volume-id 8430177d-52b7-4948-9c62-e195af4703df
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta Unassigning Amazon EBS Volumes nella AWS OpsWorks User Guide.

- Per API i dettagli, consulta AWS CLI Command [UnassignVolume](#) Reference.

## update-app

Il seguente esempio di codice mostra come utilizzare `update-app`.

### AWS CLI

Per aggiornare un'app

L'esempio seguente aggiorna un'app specificata per cambiarne il nome.

```
aws opsworks --region us-east-1 update-app --app-id 26a61ead-d201-47e3-b55c-2a7c666942f8 --name NewAppName
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta [Modifica delle app](#) nella Guida AWS OpsWorks per l'utente.

- Per API i dettagli, vedere [UpdateApp](#) in AWS CLI Command Reference.

## update-elastic-ip

Il seguente esempio di codice mostra come utilizzare `update-elastic-ip`.

### AWS CLI

Per aggiornare il nome di un indirizzo IP elastico

L'esempio seguente aggiorna il nome di un indirizzo IP elastico specificato.

```
aws opsworks --region us-east-1 update-elastic-ip --elastic-ip 54.148.130.96 --name NewIPName
```

Output: Nessuno.

Ulteriori informazioni

Per ulteriori informazioni, consulta la sezione [Gestione delle risorse](#) nella Guida AWS OpsWorks per l'utente.



- Per API i dettagli, vedere [UpdateElasticIp](#) in AWS CLI Command Reference.

## update-instance

Il seguente esempio di codice mostra come utilizzare `update-instance`.

### AWS CLI

Per aggiornare un'istanza

L'esempio seguente aggiorna il tipo di istanza specificata.

```
aws opsworks --region us-east-1 update-instance --instance-id dfc18b02-5327-493d-91a4-c5c0c448927f --instance-type c3.xlarge
```

Output: Nessuno.

Ulteriori informazioni

Per ulteriori informazioni, vedere [Modifica della configurazione dell'istanza](#) nella Guida AWS OpsWorks per l'utente.

- Per API i dettagli, vedere [UpdateInstance](#) in AWS CLI Command Reference.

## update-layer

Il seguente esempio di codice mostra come utilizzare `update-layer`.

### AWS CLI

Per aggiornare un livello

L'esempio seguente aggiorna un layer specificato per utilizzare istanze EBS ottimizzate per Amazon.

```
aws opsworks --region us-east-1 update-layer --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4 --use-efs-optimized-instances
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, vedere Modifica della configurazione di un OpsWorks livello nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [UpdateLayer](#) in AWS CLI Command Reference.

## update-my-user-profile

Il seguente esempio di codice mostra come utilizzare `update-my-user-profile`.

### AWS CLI

Per aggiornare il profilo di un utente

L'esempio seguente aggiorna il profilo dell'`development` utente per utilizzare una chiave SSH pubblica specificata. Le AWS credenziali dell'utente sono rappresentate dal `development` profilo nel `credentials` file (`~\.aws\credentials`) e la chiave si trova in un `.pem` file nella directory di lavoro.

```
aws opsworks --region us-east-1 --profile development update-my-user-profile --ssh-public-key file://development_key.pem
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, vedere Modifica delle impostazioni AWS OpsWorks utente nella Guida AWS OpsWorks per l'utente.

- Per API i dettagli, vedere [UpdateMyUserProfile](#) in AWS CLI Command Reference.

## update-rds-db-instance

Il seguente esempio di codice mostra come utilizzare `update-rds-db-instance`.

### AWS CLI

Per aggiornare un'istanza Amazon RDS DB registrata

L'esempio seguente aggiorna il valore della password principale di un'RDSistanza Amazon. Tieni presente che questo comando non modifica la password principale dell'RDSistanza, ma solo la password che fornisci a AWS OpsWorks. Se questa password non corrisponde alla password dell'RDSistanza, l'applicazione non sarà in grado di connettersi al database.

```
aws opsworks --region us-east-1 update-rds-db-instance --db-password 123456789
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta [Registrazione di RDS istanze Amazon con uno stack nella Guida per l'AWS OpsWorks utente](#).

- Per API i dettagli, consulta [Command UpdateRdsDbInstanceReference AWS CLI](#).

## update-volume

Il seguente esempio di codice mostra come utilizzare `update-volume`.

AWS CLI

Per aggiornare un volume registrato

L'esempio seguente aggiorna il punto di montaggio di un volume Amazon Elastic Block Store (AmazonEBS) registrato. Il volume è identificato dall'ID del volume, che è GUID quello AWS OpsWorks assegnato al volume quando lo registri in uno stack, non dall'ID del volume Amazon Elastic Compute Cloud (AmazonEC2). :

```
aws opsworks --region us-east-1 update-volume --volume-id 8430177d-52b7-4948-9c62-e195af4703df --mount-point /mnt/myvol
```

Uscita: nessuna.

Ulteriori informazioni

Per ulteriori informazioni, consulta [Assigning Amazon EBS Volumes to an Instance nella AWS OpsWorks User Guide](#).

- Per API i dettagli, consulta [UpdateVolume AWS CLI Command Reference](#).

## AWS OpsWorks CM esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS OpsWorks CM.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **associate-node**

Il seguente esempio di codice mostra come utilizzare `associate-node`.

AWS CLI

Per associare i nodi

Il `associate-node` comando seguente associa un nodo denominato `i-44de882p` a un server Chef Automate denominato `automate-06`, il che significa che il `automate-06` server gestisce il nodo e comunica i comandi delle ricette al nodo tramite il software `chef-client` agente installato sul nodo dal comando `associate-node`. I nomi di nodo validi sono istanze 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'"
```

L'output restituito dal comando è simile al seguente. Output:

```
{
  "NodeAssociationStatusToken": "AHUY8wFe4pdXtZC5DiJa5S0Lp5o14DH//
rHRqHDWXxwVoNBxcEy4V7R0N0Fymh7E/1Hum0BPsemPQFE6dcGaiFk"
}
```

Ulteriori informazioni

Per ulteriori informazioni, consulta [Aggiungere nodi automaticamente in Chef Automate nella Guida AWS OpsWorks per l'AWS OpsWorks utente](#).

- Per API i dettagli, vedere [AssociateNode](#) in AWS CLI Command Reference.

## create-backup

Il seguente esempio di codice mostra come utilizzare `create-backup`.

### AWS CLI

Per creare backup

Il `create-backup` comando seguente avvia un backup manuale di un server Chef Automate denominato `automate-06` nella `us-east-1` regione. Il comando aggiunge un messaggio descrittivo al backup nel `--description` parametro.

```
aws opsworks-cm create-backup \  
  --server-name 'automate-06' \  
  --description "state of my infrastructure at launch"
```

L'output mostra informazioni simili alle seguenti sul nuovo backup.

Output:

```
{  
  "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"
    }
],
}

```

Per ulteriori informazioni, consulta Backup and Restore an AWS OpsWorks for Chef Automate Server nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [CreateBackup](#) in AWS CLI Command Reference.

## create-server

Il seguente esempio di codice mostra come utilizzare `create-server`.

### AWS CLI

Per creare un server

L'`create-server` esempio seguente crea un nuovo server Chef Automate denominato `automate-06` nella regione predefinita. Tieni presente che i valori predefiniti vengono utilizzati per la maggior parte delle altre impostazioni, come il numero di backup da conservare e gli orari di avvio della manutenzione e del backup. Prima di eseguire un `create-server` comando, completa i prerequisiti in [Getting Started with AWS OpsWorks for Chef Automate nella Guida per l'utente di AWS Opsworks for Chef Automate](#).

```

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"

```

L'output mostra informazioni simili alle seguenti sul nuovo server:

```
{
  "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" ]
  }
}
```

Per ulteriori informazioni, vedere il API riferimento [UpdateServer](#) AWS OpsWorks a Chef Automate.

- Per API i dettagli, vedere [CreateServer](#) in AWS CLI Command Reference.

## delete-backup

Il seguente esempio di codice mostra come utilizzare `delete-backup`.

## AWS CLI

Per eliminare i backup

Il `delete-backup` comando seguente elimina un backup manuale o automatico di un server Chef Automate, identificato dall'ID di backup. Questo comando è utile quando ci si avvicina al numero massimo di backup che è possibile salvare o se si desidera ridurre al minimo i costi di storage di Amazon S3. :

```
aws opsworks-cm delete-backup --backup-id "automate-06-2016-11-19T23:42:40.240Z"
```

L'output mostra se l'eliminazione del backup è riuscita.

Ulteriori informazioni

Per ulteriori informazioni, consulta Backup and Restore an AWS OpsWorks for Chef Automate Server nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DeleteBackup](#) in AWS CLI Command Reference.

## `delete-server`

Il seguente esempio di codice mostra come utilizzare `delete-server`.

## AWS CLI

Per eliminare i server

Il `delete-server` comando seguente elimina un server Chef Automate, identificato dal nome del server. Dopo l'eliminazione, il server non viene più restituito dalle `DescribeServer` richieste. :

```
aws opsworks-cm delete-server --server-name "automate-06"
```

L'output mostra se l'eliminazione del server è riuscita.

Ulteriori informazioni

Per ulteriori informazioni, consulta Delete an AWS OpsWorks for Chef Automate Server nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DeleteServer](#) in AWS CLI Command Reference.



## describe-account-attributes

Il seguente esempio di codice mostra come utilizzare `describe-account-attributes`.

### AWS CLI

Per descrivere gli attributi dell'account

Il `describe-account-attributes` comando seguente restituisce informazioni sull'utilizzo da parte del tuo account delle AWS OpsWorks risorse Chef Automate. :

```
aws opsworks-cm describe-account-attributes
```

L'output per ogni voce di attributo dell'account restituita dal comando è simile al seguente. Output:

```
{
  "Attributes": [
    {
      "Maximum": 5,
      "Name": "ServerLimit",
      "Used": 2
    }
  ]
}
```

### Ulteriori informazioni

Per ulteriori informazioni, vedere il API riferimento `DescribeAccountAttributes` AWS OpsWorks a Chef Automate.

- Per API i dettagli, vedere [DescribeAccountAttributes](#) in AWS CLI Command Reference.

## describe-backups

Il seguente esempio di codice mostra come utilizzare `describe-backups`.

### AWS CLI

Per descrivere i backup

Il `describe-backups` comando seguente restituisce informazioni su tutti i backup associati all'account nella regione predefinita.

**aws opsworks-cm describe-backups**

L'output per ogni voce di backup restituita dal comando è simile al seguente.

Output:

```
{
  "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"
    }
  ],
}
```

Per ulteriori informazioni, consulta Backup and Restore an AWS OpsWorks for Chef Automate Server nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeBackups](#) in AWS CLI Command Reference.

## describe-events

Il seguente esempio di codice mostra come utilizzare `describe-events`.

### AWS CLI

Per descrivere gli eventi

L'`describe-events` esempio seguente restituisce informazioni su tutti gli eventi associati al server Chef Automate specificato.

```
aws opsworks-cm describe-events \
  --server-name 'automate-06'
```

L'output per ogni voce di evento restituita dal comando è simile al seguente esempio:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Suggerimenti generali per la risoluzione dei problemi](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DescribeEvents](#) in AWS CLI Command Reference.

## describe-node-association-status

Il seguente esempio di codice mostra come utilizzare `describe-node-association-status`.

### AWS CLI

Per descrivere lo stato dell'associazione dei nodi

Il `describe-node-association-status` comando seguente restituisce lo stato di una richiesta di associazione di un nodo a un server Chef Automate denominato `automate-06`.

```
aws opsworks-cm describe-node-association-status --server-  
name "automate-06" --node-association-status-token "AflJKl+/  
GoKLZJBdDQEx0065CDi57b1Qe9nKM8joSok0pQ9xr8DqApBN9/106sLdSvlfDEKkEx+eoCHvjowHa0s="
```

L'output per ogni voce di attributo dell'account restituita dal comando è simile al seguente. Output:

```
{  
  "NodeAssociationStatus": "IN_PROGRESS"  
}
```

### Ulteriori informazioni

Per ulteriori informazioni, vedere il API riferimento DescribeNodeAssociationStatus AWS OpsWorks a Chef Automate.

- Per API i dettagli, vedere [DescribeNodeAssociationStatus](#) in AWS CLI Command Reference.

## describe-servers

Il seguente esempio di codice mostra come utilizzare `describe-servers`.

### AWS CLI

Per descrivere i server

Il `describe-servers` comando seguente restituisce informazioni su tutti i server associati all'account e nella regione predefinita. :

```
aws opsworks-cm describe-servers
```

L'output per ogni voce del server restituita dal comando è simile al seguente. Output:

```
{  
  "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" ]
}
]
}

```

## Ulteriori informazioni

Per ulteriori informazioni, `DescribeServers` consulta la API Guida AWS OpsWorks di Chef Automate.

- Per API i dettagli, vedere [DescribeServers](#) in AWS CLI Command Reference.

## disassociate-node

Il seguente esempio di codice mostra come utilizzare `disassociate-node`.

### AWS CLI

Per dissociare i nodi

Il `disassociate-node` comando seguente dissocia un nodo denominato `i-44de882p`, rimuovendo il nodo dalla gestione da parte di un server Chef Automate denominato `automate-06`. I nomi di nodo validi sono istanze EC2 IDs :

```
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'"
```

L'output restituito dal comando è simile al seguente. Output:

```
{  
  "NodeAssociationStatusToken": "AHUY8wFe4pdXtZC5DiJa5S0Lp5o14DH//  
rHRqHDWxwVoNBxcEy4V7R0N0Fymh7E/1Hum0BPsemPQFE6dcGaiFk"  
}
```

Ulteriori informazioni

Per ulteriori informazioni, consulta [Delete an AWS OpsWorks for Chef Automate Server](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [DisassociateNode](#) in AWS CLI Command Reference.

## restore-server

Il seguente esempio di codice mostra come utilizzare `restore-server`.

### AWS CLI

Per ripristinare un server

Il `restore-server` comando seguente esegue un ripristino sul posto di un server Chef Automate denominato `automate-06` nella regione predefinita da un backup con un ID di `automate-06-2016-11-22T16:13:27.998Z`. Il ripristino di un server ripristina le connessioni ai nodi gestiti dal server Chef Automate al momento dell'esecuzione del backup specificato.

```
aws opsworks-cm restore-server --backup-id «automate-06-2016-11-22T 16:13:27.998 Z» --  
server-name «automate-06»
```

L'output è solo l'ID del comando. Output:

```
(None)
```

Ulteriori informazioni

Per ulteriori informazioni, consulta [Restore a Failed AWS OpsWorks for Chef Automate Server](#) nella Guida per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [RestoreServer](#) in AWS CLI Command Reference.

## start-maintenance

Il seguente esempio di codice mostra come utilizzare `start-maintenance`.

### AWS CLI

Per avviare la manutenzione

L'esempio seguente avvia manualmente la manutenzione sul server Chef Automate o Puppet Enterprise specificato nella regione predefinita. Questo comando è utile se un precedente tentativo di manutenzione automatica non è riuscito e la causa sottostante dell'errore di manutenzione è stata risolta.

```
aws opsworks-cm start-maintenance \  
  --server-name 'automate-06'
```

Output:

```
{  
  "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-EXAMPLEv8gjf5f.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"
}
}

```

Per ulteriori informazioni, consulta [Manutenzione del sistema \(server Puppet Enterprise\)](#) o [Manutenzione del sistema \(server Chef Automate\) nella Guida](#) per l'AWS OpsWorks utente.

- Per API i dettagli, vedere [StartMaintenance](#) in AWS CLI Command Reference.

## update-server-engine-attributes

Il seguente esempio di codice mostra come utilizzare `update-server-engine-attributes`.

### AWS CLI

Per aggiornare gli attributi del motore del server

Il `update-server-engine-attributes` comando seguente aggiorna il valore dell'attributo `CHEF_PIVOTAL_KEY` engine per un server Chef Automate denominato `automate-06`. Al momento non è possibile modificare il valore di altri attributi del motore.

```

aws opsworks-cm update-server-engine-attributes \
  --attribute-name CHEF_PIVOTAL_KEY \
  --attribute-value "new key value" \
  --server-name "automate-06"

```

L'output mostra informazioni simili alle seguenti sul server aggiornato.

```

{
  "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" ]
  }
}

```

Per ulteriori informazioni, vedere il API riferimento [UpdateServerEngineAttributes](#) AWS OpsWorks a Chef Automate.

- Per API i dettagli, vedere [UpdateServerEngineAttributes](#) in AWS CLI Command Reference.

## update-server

Il seguente esempio di codice mostra come utilizzare `update-server`.

### AWS CLI

Per aggiornare un server

Il `update-server` comando seguente aggiorna l'ora di inizio della manutenzione del server Chef Automate specificato nella regione predefinita. Il `--preferred-maintenance-window`

parametro viene aggiunto per modificare il giorno e l'ora di inizio della manutenzione del server al lunedì alle 9:15. UTC. :

```
aws opsworks-cm update-server \  
  --server-name "automate-06" \  
  --preferred-maintenance-window "Mon:09:15"
```

L'output mostra informazioni simili alle seguenti sul server aggiornato.

```
{  
  "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" ]  
  }  
}
```

Per ulteriori informazioni, vedere il API riferimento [UpdateServer](#) AWS OpsWorks a Chef Automate.

- Per API i dettagli, vedere [UpdateServer](#) in AWS CLI Command Reference.

## Organizations: esempi che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Organizations.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

### Azioni

#### **accept-handshake**

Il seguente esempio di codice mostra come utilizzare `accept-handshake`.

AWS CLI

Accettare una stretta di mano da un altro account

Bill, il proprietario di un'organizzazione, ha già invitato l'account di Juan a entrare a far parte della sua organizzazione. L'esempio seguente mostra l'account di Juan che accetta la stretta di mano e quindi accetta l'invito.

```
aws organizations accept-handshake --handshake-id h-examplehandshakeid111
```

L'output mostra quanto segue:

```
{
  "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"
  }
}

```

- Per API i dettagli, vedere [AcceptHandshake](#) in AWS CLI Command Reference.

## attach-policy

Il seguente esempio di codice mostra come utilizzare `attach-policy`.

### AWS CLI

Per allegare una policy a una root, a un'unità organizzativa o a un account

#### Esempio 1

L'esempio seguente mostra come collegare una policy di controllo del servizio (SCP) a un'unità organizzativa:

```
aws organizations attach-policy
    --policy-id p-examplepolicyid111
    --target-id ou-examplerootid111-exampleouid111
```

#### Esempio 2

L'esempio seguente mostra come allegare una policy di controllo del servizio direttamente a un account:

```
aws organizations attach-policy
    --policy-id p-examplepolicyid111
    --target-id 333333333333
```

- Per API i dettagli, vedere [AttachPolicy](#) in AWS CLI Command Reference.

## cancel-handshake

Il seguente esempio di codice mostra come utilizzare `cancel-handshake`.

### AWS CLI

Per annullare una stretta di mano inviata da un altro account

In precedenza Bill aveva inviato un invito all'account di Susan per entrare a far parte della sua organizzazione. Cambia idea e decide di annullare l'invito prima che Susan lo accetti. L'esempio seguente mostra l'annullamento di Bill:

```
aws organizations cancel-handshake --handshake-id h-examplehandshakeid111
```

L'output include un oggetto handshake che mostra che lo stato è ora CANCELED:

```
{
  "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
}
}

```

- Per API i dettagli, vedere [CancelHandshake](#) in AWS CLI Command Reference.

## create-account

Il seguente esempio di codice mostra come utilizzare `create-account`.

### AWS CLI

Per creare un account membro che faccia automaticamente parte dell'organizzazione

L'esempio seguente mostra come creare un account membro in un'organizzazione.

L'account membro è configurato con il nome Account di produzione e l'indirizzo e-mail `susan@example.com`. Organizations crea automaticamente un IAM ruolo utilizzando il nome predefinito di `OrganizationAccountAccessRole` perché il `roleName` parametro non è specificato. Inoltre, l'impostazione che consente IAM agli utenti o ai ruoli con autorizzazioni sufficienti di accedere ai dati di fatturazione dell'account viene impostata sul valore predefinito di `ALLOW` perché il `IamUserAccessToBilling` parametro non è specificato. Organizations invia automaticamente a Susan un'e-mail di «Benvenuto a AWS»:

```
aws organizations create-account --email susan@example.com --account-
name "Production Account"
```

L'output include un oggetto di richiesta che mostra che lo stato è ora `IN_PROGRESS`:

```

{
  "CreateAccountStatus": {
    "State": "IN_PROGRESS",
    "Id": "car-examplecreateaccountrequestid111"
  }
}

```

Successivamente è possibile interrogare lo stato corrente della richiesta fornendo il valore di risposta `Id` al `describe-create-account-status` comando come valore per il `create-account-request-id` parametro.

Per ulteriori informazioni, consulta [Creare un AWS account nella tua organizzazione nella AWS Organizations Users Guide](#).

- Per API i dettagli, vedere [CreateAccount](#) in AWS CLI Command Reference.

## create-organization

Il seguente esempio di codice mostra come utilizzare `create-organization`.

### AWS CLI

Esempio 1: creare una nuova organizzazione

Bill desidera creare un'organizzazione utilizzando le credenziali dell'account 1111. L'esempio seguente mostra che l'account diventa l'account principale nella nuova organizzazione. Poiché non specifica un set di funzionalità, per impostazione predefinita la nuova organizzazione prevede che tutte le funzionalità siano abilitate e le politiche di controllo del servizio sono abilitate nella directory principale.

```
aws organizations create-organization
```

L'output include un oggetto dell'organizzazione con dettagli sulla nuova organizzazione:

```
{
  "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"
    }
}

```

Esempio 2: creare una nuova organizzazione con solo le funzionalità di fatturazione consolidate abilitate

L'esempio seguente crea un'organizzazione che supporta solo le funzionalità di fatturazione consolidata:

```
aws organizations create-organization --feature-set CONSOLIDATED_BILLING
```

L'output include un oggetto dell'organizzazione con dettagli sulla nuova organizzazione:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, vedere [Creating an Organization](#) nella *AWS Organizations Users Guide*.

- Per API i dettagli, vedere [CreateOrganization](#) in *AWS CLI Command Reference*.

## create-organizational-unit

Il seguente esempio di codice mostra come utilizzare `create-organizational-unit`.

### AWS CLI

Per creare un'unità organizzativa in un'unità organizzativa principale o principale

L'esempio seguente mostra come creare un'unità organizzativa denominata AccountingOU:

```
aws organizations create-organizational-unit --parent-id r-examplerootid111 --
name AccountingOU
```

L'output include un organizationalUnit oggetto con dettagli sulla nuova unità organizzativa:

```
{
  "OrganizationalUnit": {
    "Id": "ou-examplerootid111-exampleoid111",
    "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-
    exemplerooid111-exampleoid111",
    "Name": "AccountingOU"
  }
}
```

- Per API i dettagli, vedere [CreateOrganizationalUnit](#) in AWS CLI Command Reference.

## create-policy

Il seguente esempio di codice mostra come utilizzare `create-policy`.

### AWS CLI

Esempio 1: creare una politica con un file sorgente di testo per la JSON politica

L'esempio seguente mostra come creare una policy di controllo del servizio (SCP) denominata `AllowAllS3Actions`. Il contenuto della policy è tratto da un file sul computer locale chiamato `policy.json`.

```
aws organizations create-policy --content file://policy.json --
name AllowAllS3Actions, --type SERVICE_CONTROL_POLICY --description "Allows
delegation of all S3 actions"
```

L'output include un oggetto di policy con dettagli sulla nuova policy:

```
{
  "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"
    }
}

```

Esempio 2: creare una politica con una JSON politica come parametro

L'esempio seguente mostra come creare la stessa SCP, questa volta incorporando il contenuto della policy come JSON stringa nel parametro. La stringa deve essere cancellata con barre rovesciate prima delle virgolette doppie per garantire che vengano trattate come valori letterali nel parametro, che a sua volta è circondato da virgolette doppie:

```

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

```

Per ulteriori informazioni sulla creazione e l'utilizzo delle politiche nell'organizzazione, vedere [Managing Organization Policies](#) nella *AWS Organizations User Guide*.

- Per API i dettagli, vedere [CreatePolicy](#) in *AWS CLI Command Reference*.

## decline-handshake

Il seguente esempio di codice mostra come utilizzare `decline-handshake`.

### AWS CLI

Per rifiutare una stretta di mano inviata da un altro account

L'esempio seguente mostra che Susan, un'amministratore proprietaria dell'account 222222222222, rifiuta un invito a far parte dell'organizzazione di Bill. L' `DeclineHandshake` operazione restituisce un oggetto handshake, che mostra che lo stato attuale è: `DECLINED`

```

aws organizations decline-handshake --handshake-id h-examplehandshakeid111

```

L'output include un oggetto handshake che mostra il nuovo stato di: `DECLINED`

```

{
    "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"
}
```

```
}
```

- Per API i dettagli, vedere [DeclineHandshake](#) in AWS CLI Command Reference.

## delete-organization

Il seguente esempio di codice mostra come utilizzare `delete-organization`.

### AWS CLI

Per eliminare un'organizzazione

L'esempio seguente mostra come eliminare un'organizzazione. Per eseguire questa operazione, devi essere un amministratore dell'account principale dell'organizzazione. L'esempio presuppone che in precedenza siano stati rimossi tutti gli account dei membri e le politiche dall'organizzazione: OUs

```
aws organizations delete-organization
```

- Per API i dettagli, vedere [DeleteOrganization](#) in AWS CLI Command Reference.

## delete-organizational-unit

Il seguente esempio di codice mostra come utilizzare `delete-organizational-unit`.

### AWS CLI

Per eliminare un'unità organizzativa

L'esempio seguente mostra come eliminare un'OU. L'esempio presuppone che in precedenza siano stati rimossi tutti gli account e altri account OUs dall'unità organizzativa:

```
aws organizations delete-organizational-unit --organizational-unit-id ou-  
examplerootid111-exampleouid111
```

- Per API i dettagli, vedere [DeleteOrganizationalUnit](#) in AWS CLI Command Reference.

## delete-policy

Il seguente esempio di codice mostra come utilizzare `delete-policy`.

## AWS CLI

Per eliminare una politica

L'esempio seguente mostra come eliminare una politica da un'organizzazione. L'esempio presuppone che in precedenza sia stata scollegata la politica da tutte le entità:

```
aws organizations delete-policy --policy-id p-examplepolicyid111
```

- Per API i dettagli, vedere [DeletePolicy](#) in AWS CLI Command Reference.

## describe-account

Il seguente esempio di codice mostra come utilizzare `describe-account`.

### AWS CLI

Per ottenere i dettagli su un account

L'esempio seguente mostra come richiedere dettagli su un account:

```
aws organizations describe-account --account-id 555555555555
```

L'output mostra un oggetto account con i dettagli sull'account:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [DescribeAccount](#) in AWS CLI Command Reference.

## describe-create-account-status

Il seguente esempio di codice mostra come utilizzare `describe-create-account-status`.

### AWS CLI

Per ottenere lo stato più recente di una richiesta di creazione di un account

L'esempio seguente mostra come richiedere lo stato più recente di una precedente richiesta di creazione di un account in un'organizzazione. Il `--request-id` specificato proviene dalla risposta della chiamata originale a `create-account`. La richiesta di creazione dell'account indica, tramite il campo di stato, che Organizations ha completato con successo la creazione dell'account.

Comando:

```
aws organizations describe-create-account-status --create-account-request-id car-examplecreateaccountrequestid111
```

Output:

```
{
  "CreateAccountStatus": {
    "State": "SUCCEEDED",
    "AccountId": "555555555555",
    "AccountName": "Beta account",
    "RequestedTimestamp": 1470684478.687,
    "CompletedTimestamp": 1470684532.472,
    "Id": "car-examplecreateaccountrequestid111"
  }
}
```

- Per API i dettagli, vedere [DescribeCreateAccountStatus](#) in AWS CLI Command Reference.

## describe-handshake

Il seguente esempio di codice mostra come utilizzare `describe-handshake`.

### AWS CLI

Per ottenere informazioni su una stretta di mano

L'esempio seguente mostra come richiedere dettagli su una stretta di mano. L'ID dell'handshake proviene dalla chiamata originale a `InviteAccountToOrganization` o da una chiamata a `ListHandshakesForAccount` o `ListHandshakesForOrganization`

```
aws organizations describe-handshake --handshake-id h-examplehandshakeid111
```

L'output include un oggetto handshake che contiene tutti i dettagli sull'handshake richiesto:

```
{
  "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"  
    }  
}
```

- Per API i dettagli, vedere [DescribeHandshake](#) in AWS CLI Command Reference.

## describe-organization

Il seguente esempio di codice mostra come utilizzare `describe-organization`.

### AWS CLI

Per ottenere informazioni sull'organizzazione attuale

L'esempio seguente mostra come richiedere dettagli su un'organizzazione:

```
aws organizations describe-organization
```

L'output include un oggetto dell'organizzazione che contiene i dettagli sull'organizzazione:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

- Per API i dettagli, vedere [DescribeOrganization](#) in AWS CLI Command Reference.

## describe-organizational-unit

Il seguente esempio di codice mostra come utilizzare `describe-organizational-unit`.

### AWS CLI

Per ottenere informazioni su un'unità organizzativa

L'esempio seguente richiede dettagli su un'unità organizzativa.

```
aws organizations describe-organizational-unit \  
  --organizational-unit-id ou-examplerootid111-exampleoid111
```

Output:

```
{  
  "OrganizationalUnit": {  
    "Name": "Accounting Group",  
    "Arn": "arn:aws:organizations::123456789012:ou/o-exampleorgid/ou-  
  exemplerooid111-exampleoid111",  
    "Id": "ou-examplerootid111-exampleoid111"  
  }  
}
```

- Per API i dettagli, vedere [DescribeOrganizationalUnit](#) in AWS CLI Command Reference.

## describe-policy

Il seguente esempio di codice mostra come utilizzare `describe-policy`.

### AWS CLI

Per ottenere informazioni su una politica

L'esempio seguente mostra come richiedere informazioni su una politica:

```
aws organizations describe-policy --policy-id p-examplepolicyid111
```

L'output include un oggetto politico che contiene dettagli sulla politica:

```
{
  "Policy": {
    "Content": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [\n    {\n      \"Effect\": \"Allow\",\n      \"Action\": \"*\",\n      \"Resource\": \"*\"\n    }\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"
    }
  }
}
```

- Per API i dettagli, vedere [DescribePolicy](#) in AWS CLI Command Reference.

## detach-policy

Il seguente esempio di codice mostra come utilizzare `detach-policy`.

### AWS CLI

Per scollegare una policy da una root, un'unità organizzativa o un account

L'esempio seguente mostra come scollegare una policy da un'unità organizzativa:

```
aws organizations detach-policy --target-id ou-examplerootid111-exampleoid111 --policy-id p-examplepolicyid111
```

- Per API i dettagli, vedere [DetachPolicy](#) in AWS CLI Command Reference.

## disable-policy-type

Il seguente esempio di codice mostra come utilizzare `disable-policy-type`.

## AWS CLI

Per disabilitare un tipo di policy in una directory principale

L'esempio seguente mostra come disabilitare il tipo di policy service control policy (SCP) in una cartella root:

```
aws organizations disable-policy-type --root-id r-examplerootid111 --policy-type SERVICE_CONTROL_POLICY
```

L'output mostra che l'elemento di PolicyTypes risposta non include più SERVICE \_ CONTROL \_POLICY:

```
{
  "Root": {
    "PolicyTypes": [],
    "Name": "Root",
    "Id": "r-examplerootid111",
    "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-examplerootid111"
  }
}
```

- Per API i dettagli, vedere [DisablePolicyType](#) in AWS CLI Command Reference.

## enable-all-features

Il seguente esempio di codice mostra come utilizzare `enable-all-features`.

## AWS CLI

Per abilitare tutte le funzionalità in un'organizzazione

Questo esempio mostra l'amministratore che chiede a tutti gli account invitati dell'organizzazione di approvare l'attivazione di tutte le funzionalità dell'organizzazione. AWS Organizations invia un'e-mail all'indirizzo registrato con ogni account membro invitato chiedendo al proprietario di approvare la modifica a tutte le funzionalità accettando l'handshake che viene inviato. Dopo che tutti gli account dei membri invitati hanno accettato l'handshake, l'amministratore dell'organizzazione può finalizzare la modifica a tutte le funzionalità e chi dispone delle autorizzazioni appropriate può creare politiche e applicarle a root e account: OUs

**aws organizations enable-all-features**

L'output è un oggetto handshake che viene inviato a tutti gli account dei membri invitati per l'approvazione:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [EnableAllFeatures](#) in AWS CLI Command Reference.

**enable-policy-type**

Il seguente esempio di codice mostra come utilizzare `enable-policy-type`.

**AWS CLI**

Per abilitare l'uso di un tipo di policy in una cartella root

L'esempio seguente mostra come abilitare il tipo di policy service control policy (SCP) in una cartella root:

```
aws organizations enable-policy-type --root-id r-examplerootid111 --policy-  
type SERVICE_CONTROL_POLICY
```

L'output mostra un oggetto root con un elemento di policyTypes risposta che mostra che ora SCPs sono abilitati:

```
{
  "Root": {
    "PolicyTypes": [
      {
        "Status": "ENABLED",
        "Type": "SERVICE_CONTROL_POLICY"
      }
    ],
    "Id": "r-examplerootid111",
    "Name": "Root",
    "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-  
examplerootid111"
  }
}
```

- Per API i dettagli, vedete [EnablePolicyType](#) in AWS CLI Command Reference.

## invite-account-to-organization

Il seguente esempio di codice mostra come utilizzare `invite-account-to-organization`.

### AWS CLI

Per invitare un account a entrare a far parte di un'organizzazione

L'esempio seguente mostra l'account principale di proprietà di `bill@example.com` che invita l'account di proprietà di `juan@example.com` a entrare a far parte di un'organizzazione:

```
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."
```

L'output include una struttura di handshake che mostra ciò che viene inviato all'account invitato:

```
{
```

```
    "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"
    }
  }
```

```
}
```

- Per API i dettagli, vedere [InviteAccountToOrganization](#) in AWS CLI Command Reference.

## leave-organization

Il seguente esempio di codice mostra come utilizzare `leave-organization`.

### AWS CLI

Lasciare un'organizzazione come account membro

L'esempio seguente mostra l'amministratore di un account membro che richiede di lasciare l'organizzazione di cui è attualmente membro:

```
aws organizations leave-organization
```

- Per API i dettagli, vedere [LeaveOrganization](#) in AWS CLI Command Reference.

## list-accounts-for-parent

Il seguente esempio di codice mostra come utilizzare `list-accounts-for-parent`.

### AWS CLI

Per recuperare un elenco di tutti gli account in una directory principale o unità organizzativa principale specificata

L'esempio seguente mostra come richiedere un elenco degli account in un'unità organizzativa:

```
aws organizations list-accounts-for-parent --parent-id ou-examplerootid111-exampleouid111
```

L'output include un elenco di oggetti di riepilogo degli account.

```
{
  "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"
    }
]
}

```

- Per API i dettagli, vedere [ListAccountsForParent](#) in AWS CLI Command Reference.

## list-accounts

Il seguente esempio di codice mostra come utilizzare `list-accounts`.

### AWS CLI

Per recuperare un elenco di tutti gli account di un'organizzazione

L'esempio seguente mostra come richiedere un elenco degli account di un'organizzazione:

```
aws organizations list-accounts
```

L'output include un elenco di oggetti di riepilogo degli account.

```

{
    "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/444444444444",
        "JoinedMethod": "INVITED",
        "JoinedTimestamp": 1481835812.143,
        "Id": "444444444444",
        "Name": "Test Account",
        "Email": "anika@example.com",
        "Status": "ACTIVE"
    }
]
}

```

- Per API i dettagli, vedere [ListAccounts](#) in AWS CLI Command Reference.

## list-children

Il seguente esempio di codice mostra come utilizzare `list-children`.

## AWS CLI

Per recuperare gli account secondari e OUs di un'unità organizzativa o root principale

L'esempio seguente illustra come elencare la radice o l'unità organizzativa che contiene l'account 4444444444444:

```
aws organizations list-children --child-type ORGANIZATIONAL_UNIT --parent-id ou-examplerootid111-exampleoid111
```

L'output mostra i due figli OUs contenuti dal genitore:

```
{
  "Children": [
    {
      "Id": "ou-examplerootid111-exampleoid111",
      "Type": "ORGANIZATIONAL_UNIT"
    },
    {
      "Id": "ou-examplerootid111-exampleoid222",
      "Type": "ORGANIZATIONAL_UNIT"
    }
  ]
}
```

- Per API i dettagli, vedere [ListChildren](#) in AWS CLI Command Reference.

## list-create-account-status

Il seguente esempio di codice mostra come utilizzare `list-create-account-status`.

### AWS CLI

Esempio 1: per recuperare un elenco delle richieste di creazione di account effettuate nell'organizzazione corrente

L'esempio seguente mostra come richiedere un elenco di richieste di creazione di account per un'organizzazione che sono state completate con successo:

```
aws organizations list-create-account-status --states SUCCEEDED
```

L'output include una serie di oggetti con informazioni su ogni richiesta.

```
{
  "CreateAccountStatuses": [
    {
      "AccountId": "4444444444444444",
      "AccountName": "Developer Test Account",
      "CompletedTimeStamp": 1481835812.143,
      "Id": "car-examplecreateaccountrequestid111",
      "RequestedTimeStamp": 1481829432.531,
      "State": "SUCCEEDED"
    }
  ]
}
```

Esempio 2: per recuperare un elenco delle richieste di creazione di account in corso effettuate nell'organizzazione corrente

L'esempio seguente ottiene un elenco di richieste di creazione di account in corso per un'organizzazione:

```
aws organizations list-create-account-status --states IN_PROGRESS
```

L'output include una serie di oggetti con informazioni su ogni richiesta.

```
{
  "CreateAccountStatuses": [
    {
      "State": "IN_PROGRESS",
      "Id": "car-examplecreateaccountrequestid111",
      "RequestedTimeStamp": 1481829432.531,
      "AccountName": "Production Account"
    }
  ]
}
```

- Per API i dettagli, vedere [ListCreateAccountStatus](#) in AWS CLI Command Reference.

## **list-handshakes-for-account**

Il seguente esempio di codice mostra come utilizzare `list-handshakes-for-account`.

## AWS CLI

Per recuperare un elenco delle strette di mano inviate a un account

L'esempio seguente mostra come ottenere un elenco di tutti gli handshake associati all'account delle credenziali utilizzate per chiamare l'operazione:

```
aws organizations list-handshakes-for-account
```

L'output include un elenco di strutture di handshake con informazioni su ogni handshake, incluso lo stato corrente:

```
{
  "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"
}
}

```

- Per API i dettagli, vedere [ListHandshakesForAccount](#) in AWS CLI Command Reference.

## list-handshakes-for-organization

Il seguente esempio di codice mostra come utilizzare `list-handshakes-for-organization`.

### AWS CLI

Per recuperare un elenco delle strette di mano associate a un'organizzazione

L'esempio seguente mostra come ottenere un elenco di handshake associati all'organizzazione corrente:

```
aws organizations list-handshakes-for-organization
```

L'output mostra due strette di mano. Il primo è un invito all'account di Juan e mostra uno stato di OPEN Il secondo è un invito all'account di Anika e mostra uno stato diACCEPTED:

```

{
  "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"
                }
            ],
            "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."
      }
    ]
  }
]
```



```
}
```

- Per API i dettagli, vedere [ListHandshakesForOrganization](#) in AWS CLI Command Reference.

## list-organizational-units-for-parent

Il seguente esempio di codice mostra come utilizzare `list-organizational-units-for-parent`.

### AWS CLI

Per recuperare un elenco di file OUs in un'unità organizzativa o root principale

L'esempio seguente mostra come ottenere un elenco di OUs in una radice specificata:

```
aws organizations list-organizational-units-for-parent --parent-id r-examplerootid111
```

L'output mostra che la radice specificata ne contiene due OUs e mostra i dettagli di ciascuna:

```
{
  "OrganizationalUnits": [
    {
      "Name": "AccountingDepartment",
      "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-examplerootid111/ou-examplerootid111-exampleouid111"
    },
    {
      "Name": "ProductionDepartment",
      "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-examplerootid111/ou-examplerootid111-exampleouid222"
    }
  ]
}
```

- Per API i dettagli, vedere [ListOrganizationalUnitsForParent](#) in AWS CLI Command Reference.

## list-parents

Il seguente esempio di codice mostra come utilizzare `list-parents`.

## AWS CLI

Per elencare il genitore OUs o i root di un account o di un'unità organizzativa secondaria

L'esempio seguente illustra come elencare l'unità organizzativa principale o principale che contiene l'account 444444444444:

```
aws organizations list-parents --child-id 444444444444
```

L'output mostra che l'account specificato si trova nell'unità organizzativa con l'ID specificato:

```
{
  "Parents": [
    {
      "Id": "ou-examplerootid111-exampleouid111",
      "Type": "ORGANIZATIONAL_UNIT"
    }
  ]
}
```

- Per API i dettagli, vedere [ListParents](#) in AWS CLI Command Reference.

## list-policies-for-target

Il seguente esempio di codice mostra come utilizzare `list-policies-for-target`.

## AWS CLI

Per recuperare un elenco degli SCPs allegati direttamente a un account

L'esempio seguente mostra come ottenere un elenco di tutte le politiche di controllo del servizio (SCPs), come specificato dal parametro `Filter`, che sono direttamente collegate a un account:

```
aws organizations list-policies-for-target --filter SERVICE_CONTROL_POLICY --target-id 444444444444
```

L'output include un elenco di strutture politiche con informazioni di riepilogo sulle politiche.

L'elenco non include le politiche che si applicano all'account a causa dell'ereditarietà dalla sua posizione in una gerarchia di unità organizzative:

```
{
```

```

    "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."
      }
    ]
  }
}

```

- Per API i dettagli, vedere [ListPoliciesForTarget](#) in AWS CLI Command Reference.

## list-policies

Il seguente esempio di codice mostra come utilizzare `list-policies`.

### AWS CLI

Per recuperare un elenco di tutte le politiche in un'organizzazione di un determinato tipo

L'esempio seguente mostra come ottenere un elenco di SCPs, come specificato dal parametro `filter`:

```
aws organizations list-policies --filter SERVICE_CONTROL_POLICY
```

L'output include un elenco di politiche con informazioni di riepilogo:

```

{
  "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"
    }
  ]
}

```

- Per API i dettagli, vedere [ListPolicies](#) in AWS CLI Command Reference.

## list-roots

Il seguente esempio di codice mostra come utilizzare `list-roots`.

### AWS CLI

Per recuperare un elenco delle radici di un'organizzazione

Questo esempio mostra come ottenere l'elenco delle radici di un'organizzazione:

```
aws organizations list-roots
```

L'output include un elenco di strutture principali con informazioni di riepilogo:

```

{
  "Roots": [
    {

```

```

        "Name": "Root",
        "Arn": "arn:aws:organizations::111111111111:root/o-
exampleorgid/r-examplerootid111",
        "Id": "r-examplerootid111",
        "PolicyTypes": [
            {
                "Status": "ENABLED",
                "Type": "SERVICE_CONTROL_POLICY"
            }
        ]
    }
]
}

```

- Per API i dettagli, vedere [ListRoots](#) in AWS CLI Command Reference.

## list-targets-for-policy

Il seguente esempio di codice mostra come utilizzare `list-targets-for-policy`.

### AWS CLI

Per recuperare un elenco delle radici e degli account a cui è associata una politica OUs

L'esempio seguente mostra come ottenere un elenco delle radici e degli account a cui è associata la politica specificata: OUs

```
aws organizations list-targets-for-policy --policy-id p-FullAWSAccess
```

L'output include un elenco di oggetti allegati con informazioni di riepilogo sulle radici e gli account a cui è associata la politica: OUs

```

{
    "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"
    }
]
}

```

- Per API i dettagli, vedere [ListTargetsForPolicy](#) in AWS CLI Command Reference.

## move-account

Il seguente esempio di codice mostra come utilizzare `move-account`.

### AWS CLI

Per spostare un account tra root o OUs

L'esempio seguente mostra come spostare l'account principale dell'organizzazione dalla root a un'unità organizzativa:

```
aws organizations move-account --account-id 333333333333 --source-parent-id r-
examplerootid111 --destination-parent-id ou-examplerootid111-exampleoid111
```

- Per API i dettagli, vedere [MoveAccount](#) in AWS CLI Command Reference.

## remove-account-from-organization

Il seguente esempio di codice mostra come utilizzare `remove-account-from-organization`.

### AWS CLI

Per rimuovere un account da un'organizzazione come account principale

L'esempio seguente mostra come rimuovere un account da un'organizzazione:

```
aws organizations remove-account-from-organization --account-id 333333333333
```

- Per API i dettagli, vedere [RemoveAccountFromOrganization](#) in AWS CLI Command Reference.

## update-organizational-unit

Il seguente esempio di codice mostra come utilizzare `update-organizational-unit`.

### AWS CLI

Per rinominare un'unità organizzativa

Questo esempio mostra come rinominare un'unità organizzativa: in questo esempio, l'unità organizzativa viene rinominata «AccountingOU»:

```
aws organizations update-organizational-unit --organizational-unit-id ou-examplerootid111-exampleoid111 --name AccountingOU
```

L'output mostra il nuovo nome:

```
{
  "OrganizationalUnit": {
    "Id": "ou-examplerootid111-exampleoid111"
    "Name": "AccountingOU",
    "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-examplerootid111-exampleoid111"
  }
}
```

- Per API i dettagli, vedere [UpdateOrganizationalUnit](#) in AWS CLI Command Reference.

## update-policy

Il seguente esempio di codice mostra come utilizzare `update-policy`.

### AWS CLI

Esempio 1: rinominare una politica





```
    "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"
    }
  }
}
```

- Per API i dettagli, vedere [UpdatePolicy](#) in AWS CLI Command Reference.

## AWS Outposts esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Outposts.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **get-outpost-instance-types**

Il seguente esempio di codice mostra come utilizzare `get-outpost-instance-types`.

#### AWS CLI

Per visualizzare i tipi di istanza su Outpost

L'esempio seguente ottiene i tipi di istanza per l'Outpost specificato.

```
aws outposts get-outpost-instance-types \  
--outpost-id op-0ab23c4567EXAMPLE
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Launch an instance on Outpost](#) nella AWS Outposts User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetOutpostInstanceTypes](#) Reference.

## get-outpost

Il seguente esempio di codice mostra come utilizzare `get-outpost`.

### AWS CLI

Per ottenere i dettagli di Outpost

L'`get-outpost` esempio seguente mostra i dettagli dell'Outpost specificato.

```
aws outposts get-outpost \  
--outpost-id op-0ab23c4567EXAMPLE
```

**Output:**

```
{
  "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": {}
  }
}
```

Per ulteriori informazioni, consulta [Lavorare con Outposts](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, consulta AWS CLI Command [GetOutpost](#)Reference.

**list-outposts**

Il seguente esempio di codice mostra come utilizzare `list-outposts`.

**AWS CLI**

Per elencare Outposts

L'`list-outposts`esempio seguente elenca gli Outposts presenti nel tuo AWS account.

```
aws outposts list-outposts
```

**Output:**

```
{
  "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": {}
  }
]
}

```

Per ulteriori informazioni, consulta [Lavorare con Outposts](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, consulta AWS CLI Command [ListOutposts](#) Reference.

## list-sites

Il seguente esempio di codice mostra come utilizzare `list-sites`.

### AWS CLI

Per elencare i siti

L'`list-sites` esempio seguente elenca i siti Outpost disponibili nel tuo AWS account.

```
aws outposts list-sites
```

Output:

```
{
  "Sites": [
    {
      "SiteId": "os-0ab12c3456EXAMPLE",
      "AccountId": "123456789012",
      "Name": "EXAMPLE",
      "Description": "example",
      "Tags": {}
    }
  ]
}
```

Per ulteriori informazioni, consulta [Lavorare con Outposts](#) nella Guida per l'utente di AWS Outposts.

- Per API i dettagli, consulta AWS CLI Command [ListSites](#)Reference.

## AWS Payment Cryptography esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Payment Cryptography.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **create-alias**

Il seguente esempio di codice mostra come utilizzare `create-alias`.

#### AWS CLI

Per creare un alias per una chiave

L'`create-alias` esempio seguente crea un alias per una chiave.

```
aws payment-cryptography create-alias \  
  --alias-name alias/sampleAlias1 \  
  --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaif1lw2h
```

Output:

```
{  
  "Alias": {  
    "AliasName": "alias/sampleAlias1",  
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/  
kwapwa6qaif1lw2h"  
  }  
}
```

Per ulteriori informazioni, vedere Informazioni [sugli alias](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateAlias](#) Reference.

## create-key

Il seguente esempio di codice mostra come utilizzare `create-key`.

### AWS CLI

Per creare una chiave

L'`create-key` esempio seguente genera una KEY TDES chiave 2 che è possibile utilizzare per generare e verificare CVV2 i valori CVV /.

```
aws payment-cryptography create-key \  
  --exportable \  
  --key-  
attributes KeyAlgorithm=TDES_2KEY, KeyUsage=TR31_C0_CARD_VERIFICATION_KEY, KeyClass=SYMMETRIC
```

Output:

```
{  
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Generazione di chiavi](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [CreateKey AWS CLI](#) Command Reference.

## delete-alias

Il seguente esempio di codice mostra come utilizzare `delete-alias`.

### AWS CLI

Per eliminare un alias

L'`delete-alias` esempio seguente elimina un alias. Non influisce sulla chiave.

```
aws payment-cryptography delete-alias \  
  --alias-name alias/sampleAlias1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Informazioni sugli alias](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteAlias](#)Reference.

## delete-key

Il seguente esempio di codice mostra come utilizzare `delete-key`.

### AWS CLI

Per eliminare una chiave

L'`delete-key` esempio seguente pianifica l'eliminazione di una chiave dopo 7 giorni, che è il periodo di attesa predefinito.

```
aws payment-cryptography delete-key \  
  --key-identifier arn:aws:payment-cryptography:us-west-2:123456789012:key/  
  kwapwa6qaifllw2h
```

Output:

```
{  
  "Key": {  
    "CreateTimestamp": "1686801198",  
    "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"
}
}

```

Per ulteriori informazioni, consulta [Eliminazione delle chiavi](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteKey](#) Reference.

## export-key

Il seguente esempio di codice mostra come utilizzare `export-key`.

### AWS CLI

Per esportare una chiave

L'`export-key` esempio seguente esporta una chiave.

```

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 }}'

```

Contenuto di `wrapping-key-certificate.pem`.

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2VENDQXFXZ0F3SUJBZ01SQU1ZZS8xMXFUK2svVz1RUDJQ0E1V
```

Output:

```
{
  "WrappedKey": {
    "KeyMaterial":
"308205A106092A864886F70D010702A08205923082058E020101310D300B06096086480165030402013082031F
    "WrappedKeyMaterialFormat": "TR34_KEY_BLOCK"
  }
}
```

Per ulteriori informazioni, [consulta Export keys](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [ExportKey AWS CLI Command Reference](#).

## get-alias

Il seguente esempio di codice mostra come utilizzare `get-alias`.

### AWS CLI

Per ottenere un alias

L'`get-alias` esempio seguente restituisce ARN la chiave associata all'alias.

```
aws payment-cryptography get-alias \
  --alias-name alias/sampleAlias1
```

Output:

```
{
  "Alias": {
    "AliasName": "alias/sampleAlias1",
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifl1w2h"
  }
}
```

Per ulteriori informazioni, consulta la sezione Informazioni [sugli alias](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetAliasReference](#).

## get-key

Il seguente esempio di codice mostra come utilizzare `get-key`.

### AWS CLI

Per ottenere i metadati di una chiave

L'`get-key` seguente restituisce i metadati della chiave associata all'`alias`. Questa operazione non restituisce materiale crittografico.

```
aws payment-cryptography get-key \  
  --key-identifier alias/sampleAlias1
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Ottieni le chiavi](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [GetKey AWS CLI Command Reference](#).

## get-parameters-for-export

Il seguente esempio di codice mostra come utilizzare `get-parameters-for-export`.

### AWS CLI

Per inizializzare il processo di esportazione

L'`get-parameters-for-export` seguente genera una coppia di chiavi, firma la chiave e quindi restituisce il certificato e la radice del certificato.

```

aws payment-cryptography get-parameters-for-export \
  --signing-key-algorithm RSA_2048 \
  --key-material-type TR34_KEY_BLOCK

```

Output:

```

{
  "ExportToken": "export-token-ep5cwyzone7oya53",
  "ParametersValidUntilTimestamp": "1687415640",
  "SigningKeyAlgorithm": "RSA_2048",
  "SigningKeyCertificate":

  "MIICiTCCAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
  VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
  b24xFDASBgNVBA5TC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmAd
  BgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
  MTIwNDI1MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
  VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC0lBTSBDb25z
  b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWVxHmAdBgkqhkiG9w0BCQEWEG5vb25lQGFT
  YXpvbi5jb20wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ

```

```

21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVvXyUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
"SigningKeyCertificateChain":
"NIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMakGA1UEBhMC
VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAStC01BTSBDb25zb2x1MRIwEAYDVQQDEwLUZXN0Q21sYWxhZAd
BgkqhkiG9w0BCQEWEG5vb25lQGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMakGA1UEBhMCMVVMxCzAJBgNVBAGTAldBMRAwDgYD
VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC01BTSBDb25z
b2x1MRIwEAYDVQQDEwLUZXN0Q21sYWxhZAdBgkqhkiG9w0BCQEWEG5vb25lQGft
YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVvXyUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}

```

Per ulteriori informazioni, [consulta Export keys](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [GetParametersForExport AWS CLI Command Reference](#).

## get-parameters-for-import

Il seguente esempio di codice mostra come utilizzare `get-parameters-for-import`.

### AWS CLI

Per inizializzare il processo di importazione

L'`get-parameters-for-import` seguente genera una coppia di chiavi, firma la chiave e quindi restituisce il certificato e la radice del certificato.

```

aws payment-cryptography get-parameters-for-import \
  --key-material-type TR34_KEY_BLOCK \
  --wrapping-key-algorithm RSA_2048

```

Output:

```
{
  "ImportToken": "import-token-qgmafpaa7nt2kfbb",
  "ParametersValidUntilTimestamp": "1687415640",
  "WrappingKeyAlgorithm": "RSA_2048",
  "WrappingKeyCertificate":
  "MIICiTCCAfICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCMC
  VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
  b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAd
  BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
  MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCMCVMxCzAJBgNVBAGTAldBMRAwDgYD
  VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
  b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAdBgkqhkiG9w0BCQEWEG5vb251QGft
  YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
  21uUSfwfEvySwTC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
  rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
  Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
  nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
  FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvJx79LjStB
  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
  "WrappingKeyCertificateChain":
  "MIICiTCCAfICCCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCMC
  VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
  b24xFDASBgNVBAwTC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAd
  BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
  MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCMCVMxCzAJBgNVBAGTAldBMRAwDgYD
  VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
  b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxhZAdBgkqhkiG9w0BCQEWEG5vb251QGft
  YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
  21uUSfwfEvySwTC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
  rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
  Ibb30hjZncvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
  nUHVvXyUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
  FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvJx79LjStB
  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}
```

Per ulteriori informazioni, consulta [Import keys](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [GetParametersForImport AWS CLI Command Reference](#).

## get-public-key-certificate

Il seguente esempio di codice mostra come utilizzare `get-public-key-certificate`.

## AWS CLI

Per restituire la chiave pubblica

L'get-public-key-certificateesempio seguente restituisce la parte di chiave pubblica di una coppia di chiavi.

```
aws payment-cryptography get-public-key-certificate \
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaiFlLw2h
```

Output:

```
{
  "KeyCertificate":
  "MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
  VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
  b24xFDASBgNVBAStC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAd
  BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
  MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
  VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC01BTSBDb25z
  b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAdBgkqhkiG9w0BCQEWEG5vb251QGft
  YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
  21uUSfwfEvySWtC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
  rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
  Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
  nUHVvxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
  FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjStB
  NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
  "KeyCertificateChain":
  "MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
  VVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
  b24xFDASBgNVBAStC01BTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAd
  BgkqhkiG9w0BCQEWEG5vb251QGftYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
  MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYD
  VQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAStC01BTSBDb25z
  b2x1MRIwEAYDVQQDEw1UZXR0Q21sYWx1eHAdBgkqhkiG9w0BCQEWEG5vb251QGft
  YXpvbi5jb20wZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
  21uUSfwfEvySWtC2XADZ4nB+BLyGVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
  rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpE
  Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
  nUHVvxYUntneD9+h8Mg9q6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
  FFBjvSfpJI1J00zbhNYS5f6GuoEDmFJ10ZxBHjJnyp3780D8uTs7fLvjx79LjStB
```

```
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}
```

Per ulteriori informazioni, consulta [Ottenerne la chiave/certificato pubblico associato a una coppia di chiavi](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta Command [GetPublicKeyCertificate](#)Reference AWS CLI .

## import-key

Il seguente esempio di codice mostra come utilizzare `import-key`.

### AWS CLI

Per importare una chiave TR-34

L'`import-key`esempio seguente importa una chiave TR-34.

```
aws payment-cryptography import-key \
  --key-material='{ "Tr34KeyBlock": {" \
    CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-
cryptography:us-west-2:123456789012:key/rmm5wn2q564njjm", \
    "ImportToken": "import-token-5ott6ho5nts7bbc", \
    "KeyBlockFormat": "X9_TR34_2012", \
    "SigningKeyCertificate": file://signing-key-certificate.pem, \
    "WrappedKeyBlock": file://wrapped-key-block.pem } }'
```

Contenuto di `signing-key-certificate.pem`.

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUV2RENDQXFTZ0F3SUJBZ01RYWVCK25IbE1WZU1PR1ZiNjU1Q2Jz
```

Contenuto di `wrapped-key-block.pem`.

```
3082059806092A864886F70D010702A082058930820585020101310D300B06096086480165030402013082031606
```

Output:

```
{
  "Key": {
    "CreateTimestamp": "2023-06-09T16:56:27.621000-07:00",
```



```
    "Enabled": true,
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
bzmvgyx dg3sktwxd",
    "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"
  }
}
```

Per ulteriori informazioni, consulta [Import keys](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [ImportKey AWS CLI Command Reference](#).

## list-aliases

Il seguente esempio di codice mostra come utilizzare `list-aliases`.

### AWS CLI

Per ottenere un elenco di alias

L'esempio seguente mostra tutti gli alias del tuo account in questa regione.

```
aws payment-cryptography list-aliases
```

**Output:**

```
{
  "Aliases": [
    {
      "AliasName": "alias/sampleAlias1",
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/kwapwa6qaif1lw2h"
    },
    {
      "AliasName": "alias/sampleAlias2",
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/kwapwa6qaif1lw2h"
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione Informazioni [sugli alias](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta AWS CLI Command [ListAliasesReference](#).

**list-keys**

Il seguente esempio di codice mostra come utilizzare `list-keys`.

**AWS CLI**

Per ottenere un elenco di chiavi

L'`list-keys` esempio seguente mostra tutte le chiavi del tuo account in questa regione.

```
aws payment-cryptography list-keys
```

**Output:**

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [List keys](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [ListKeys AWS CLI Command Reference](#).

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per ottenere l'elenco dei tag per una chiave

L'`list-tags-for-resource`esempio seguente ottiene i tag per una chiave.

```
aws payment-cryptography list-tags-for-resource \
```

```
--resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaiFlw2h
```

Output:

```
{  
  "Tags": [  
    {  
      "Key": "BIN",  
      "Value": "20151120"  
    },  
    {  
      "Key": "Project",  
      "Value": "Production"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Gestire i tag chiave con API le operazioni](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## restore-key

Il seguente esempio di codice mostra come utilizzare `restore-key`.

AWS CLI

Per ripristinare una chiave di cui è prevista l'eliminazione

L'`restore-key` esempio seguente annulla l'eliminazione di una chiave.

```
aws payment-cryptography restore-key \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
kwapwa6qaiFlw2h
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Eliminazione delle chiavi](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta AWS CLI Command [RestoreKey](#) Reference.

## start-key-usage

Il seguente esempio di codice mostra come utilizzare `start-key-usage`.

### AWS CLI

Per abilitare una chiave

L'esempio seguente consente di utilizzare una chiave.

```
aws payment-cryptography start-key-usage \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaif1lw2h
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Abilitazione e disattivazione delle chiavi](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta AWS CLI Command [StartKeyUsageReference](#).

## stop-key-usage

Il seguente esempio di codice mostra come utilizzare `stop-key-usage`.

### AWS CLI

Per disattivare una chiave

L'`stop-key-usage`esempio seguente disabilita una chiave.

```
aws payment-cryptography stop-key-usage \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaifllw2h
```

Output:

```
{  
  "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"  
  },  
  "KeyCheckValue": "369D",  
  "KeyCheckValueAlgorithm": "ANSI_X9_24",  
  "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",  
  "KeyState": "CREATE_COMPLETE",  
  "UsageStartTimestamp": "1686800690"  
}
```

```
}  
}
```

Per ulteriori informazioni, vedere [Abilitazione e disabilitazione delle chiavi](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta AWS CLI Command [StopKeyUsageReference](#).

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare una chiave

L'`tag-resource` esempio seguente etichetta una chiave.

```
aws payment-cryptography tag-resource \  
  --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaiFlLw2h \  
  --tags Key=sampleTag,Value=sampleValue
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing key tag](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [TagResource AWS CLI Command Reference](#).

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da una chiave

L'`untag-resource` esempio seguente rimuove un tag da una chiave.

```
aws payment-cryptography untag-resource \  
  --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaiFlLw2h \  
  --tag-key sampleTag
```



```
--tag-keys sampleTag
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing key tag](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [UntagResource AWS CLI Command Reference](#).

## update-alias

Il seguente esempio di codice mostra come utilizzare `update-alias`.

### AWS CLI

Per aggiornare un alias

L'`update-alias` esempio seguente associa l'alias a una chiave diversa.

```
aws payment-cryptography update-alias \  
  --alias-name alias/sampleAlias1 \  
  --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/  
tqv5yij6wtxx64pi
```

Output:

```
{  
  "Alias": {  
    "AliasName": "alias/sampleAlias1",  
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/  
tqv5yij6wtxx64pi "  
  }  
}
```

Per ulteriori informazioni, consulta [About alias](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateAlias](#) Reference.

## AWS Payment Cryptography Esempi di piani dati che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Payment Cryptography Data Plane.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **decrypt-data**

Il seguente esempio di codice mostra come utilizzare `decrypt-data`.

AWS CLI

Per decrittografare il testo cifrato

L'`decrypt-data` esempio seguente decrittografa i dati di testo cifrato utilizzando una chiave simmetrica. Per questa operazione, la chiave deve essere impostata su e impostata su. `KeyModesOfUse 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}'
```

Output:

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
  "KeyCheckValue": "71D7AE",
  "PlainText": "31323334313233343132333431323334"
}
```

Per ulteriori informazioni, [consulta `Decrittare i dati`](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta Command [DecryptData](#)Reference AWS CLI .

## encrypt-data

Il seguente esempio di codice mostra come utilizzare `encrypt-data`.

### AWS CLI

Per crittografare i dati

L'`encrypt-data`esempio seguente crittografa i dati in chiaro utilizzando una chiave simmetrica. Per questa operazione, la chiave deve essere impostata su `KeyModesOfUse` impostata su `Encrypt` `KeyUsage` `TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`

```
aws payment-cryptography-data encrypt-data \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaifl1w2h \  
  --plain-text 31323334313233343132333431323334 \  
  --encryption-attributes 'Symmetric={Mode=CBC}'
```

Output:

```
{  
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/  
  kwapwa6qaifl1w2h",  
  "KeyCheckValue": "71D7AE",  
  "CipherText": "33612AB9D6929C3A828EB6030082B2BD"  
}
```

Per ulteriori informazioni, [consulta la sezione Crittografa i dati](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta AWS CLI Command [EncryptData](#)Reference.

## generate-card-validation-data

Il seguente esempio di codice mostra come utilizzare `generate-card-validation-data`.

### AWS CLI

Per generare un CVV

L'`generate-card-validation-data` seguente genera un CVV/CVV2.

```
aws payment-cryptography-data generate-card-validation-data \
  --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
  --primary-account-number=171234567890123 \
  --generation-attributes CardVerificationValue2={CardExpiryDate=0123}
```

Output:

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
  "KeyCheckValue": "CADD1",
  "ValidationData": "801"
}
```

Per ulteriori informazioni, consulta [Generare i dati delle carte](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [GenerateCardValidationData AWS CLI Command Reference](#).

## generate-mac

Il seguente esempio di codice mostra come utilizzare `generate-mac`.

### AWS CLI

Per generare un MAC

L'`generate-card-validation-data` seguente genera un codice di autenticazione dei messaggi basato su hash (HMAC) per l'autenticazione dei dati delle carte utilizzando l'algoritmo HMAC \_ SHA256 e una chiave di HMAC crittografia. La chiave deve essere KeyUsage impostata su TR31\_M7\_HMAC\_KEY e KeyModesOfUse su 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
```

**Output:**

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaif1lw2h,
  "KeyCheckValue": "2976E7",
  "Mac": "ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C"
}
```

Per ulteriori informazioni, consulta [Generate MAC](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [GenerateMac AWS CLI Command Reference](#).

**generate-pin-data**

Il seguente esempio di codice mostra come utilizzare `generate-pin-data`.

**AWS CLI**

Per generare un PIN

L'`generate-card-validation-data` esempio seguente genera un nuovo ordine casuale PIN utilizzando lo PIN schema Visa.

```
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}
```

**Output:**

```
{
  "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"
    }
  }
}

```

Per ulteriori informazioni, consulta [Generare PIN dati](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [GeneratePinData AWS CLI Command Reference](#).

## re-encrypt-data

Il seguente esempio di codice mostra come utilizzare `re-encrypt-data`.

### AWS CLI

Per crittografare nuovamente i dati con una chiave diversa

L'`re-encrypt-data` esempio seguente decrittografa il testo cifrato che è stato crittografato utilizzando una chiave AES simmetrica e lo cripta nuovamente utilizzando una chiave derivata unica chiave per transazione (`DUKPT`).

```

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"}}'

```

Output:

```

{
  "CipherText":
"F94959DA30EEFF0C035483C6067667CF6796E3C1AD28C2B61F9CFEB772A8DD41C0D6822931E0D3B1",
  "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
jl6ythkcvzesbxen",
  "KeyCheckValue": "2E8CD9"
}

```

```
}

```

Per ulteriori informazioni, consulta la sezione Crittografia [e decrittografa](#) i dati nella Payment Cryptography User Guide.AWS

- Per i API dettagli, consulta [ReEncryptData](#) Command Reference.AWS CLI

## translate-pin-data

Il seguente esempio di codice mostra come utilizzare `translate-pin-data`.

### AWS CLI

Per tradurre PIN i dati

L'`translate-pin-data`esempio seguente traduce un blocco PIN da PEK TDES crittografia utilizzando un PIN blocco ISO 0 a un PIN blocco AES ISO 4 utilizzando l'`DUKPT`algoritmo.

```
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"
```

Output:

```
{
  "PinBlock": "1F4209C670E49F83E75CC72E81B787D9",
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt
  "KeyCheckValue": "7CC9E2"
}
```

Per ulteriori informazioni, consulta [Translate PIN data](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [TranslatePinData AWS CLI](#) Command Reference.

## verify-auth-request-ciphertext

Il seguente esempio di codice mostra come utilizzare `verify-auth-request-ciphertext`.

### AWS CLI

Per verificare una richiesta di autenticazione

L'esempio seguente verifica un Ciphertext di richiesta di autorizzazione (ARQC).

```
aws payment-cryptography-data verify-auth-request-ciphertext \
  --auth-request-ciphertext F6E1BD1E6037FB3E \
  --auth-response-attributes '{"ArpcMethod1": {"AuthResponseCode": "1111"}}' \
  --key-identifier arn:aws:payment-cryptography:us-west-2:111122223333:key/
pboipdfzd4mdklya \
  --major-key-derivation-mode "EMV_OPTION_A" \
  --session-key-derivation-attributes '{"EmvCommon":
{"ApplicationTransactionCounter": "1234", "PanSequenceNumber":
"01", "PrimaryAccountNumber": "471234567890123"}}' \
  --transaction-data "123456789ABCDEF"
```

Output:

```
{
  "AuthResponseValue": "D899B8C6FBF971AA",
  "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
pboipdfzd4mdklya",
  "KeyCheckValue": "985792"
}
```

Per ulteriori informazioni, vedete il [cittogramma Verify auth request \(ARQC\) nella AWS Payment Cryptography User Guide](#).

- Per API i dettagli, consulta Command [VerifyAuthRequestCiphertext](#) Reference AWS CLI .

## verify-card-validation-data

Il seguente esempio di codice mostra come utilizzare `verify-card-validation-data`.

### AWS CLI

Per convalidare un CVV



L'`verify-card-validation-data` seguente convalida un CVV/CVV2 per a. PAN

```
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
```

Output:

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
  tqv5yij6wtxx64pi",
  "KeyCheckValue": "CADD1"
}
```

Per ulteriori informazioni, consulta la sezione [Verifica dei dati della carta](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [VerifyCardValidationData AWS CLI Command Reference](#).

## verify-mac

Il seguente esempio di codice mostra come utilizzare `verify-mac`.

### AWS CLI

Per verificare un MAC

L'`verify-mac` seguente verifica un codice di autenticazione dei messaggi basato su hash (HMAC) per l'autenticazione dei dati delle carte utilizzando l'algoritmo HMAC \_ SHA256 e una chiave di HMAC crittografia.

```
aws payment-cryptography-data verify-mac \
  --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6 \
  --message-
  data "3b343038383439303031303733393431353d32343038323236303030373030303f33" \
  --verification-attributes='Algorithm=HMAC_SHA256' \
  --mac ED87F26E961CGD0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C
```

**Output:**

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
qno151ghrzunce6,
  "KeyCheckValue": "2976E7",
}
```

Per ulteriori informazioni, consulta [Verify MAC](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [VerifyMac AWS CLI Command Reference](#).

**verify-pin-data**

Il seguente esempio di codice mostra come utilizzare `verify-pin-data`.

**AWS CLI**

Per verificare un PIN

L'`verify-pin-data` esempio seguente convalida a PIN per aPAN.

```
aws payment-cryptography-data verify-pin-data \
  --verification-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 \
  --verification-attributes
  VisaPin="{PinVerificationKeyIndex=1,VerificationValue=5507}" \
  --encrypted-pin-block AC17DC148BDA645E
```

**Output:**

```
{
  "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",
```

```
}
```

Per ulteriori informazioni, consulta [Verify PIN data](#) nella AWS Payment Cryptography User Guide.

- Per API i dettagli, consulta [VerifyPinData AWS CLI Command Reference](#).

## Esempi di utilizzo di Amazon Pinpoint AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Pinpoint. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-app**

Il seguente esempio di codice mostra come utilizzare `create-app`.

AWS CLI

Esempio 1: creazione di un'applicazione

L'esempio `create-app` seguente crea una nuova applicazione (progetto).

```
aws pinpoint create-app \  
  --create-application-request Name=ExampleCorp
```

Output:

```
{  
  "ApplicationResponse": {
```

```

    "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
    "Id": "810c7aab86d42fb2b56c8c966example",
    "Name": "ExampleCorp",
    "tags": {}
  }
}

```

## Esempio 2: creazione di un'applicazione con tag

L'esempio `create-app` seguente crea una nuova applicazione (progetto) e associa un tag (chiave e valore) all'applicazione.

```

aws pinpoint create-app \
  --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}

```

Output:

```

{
  "ApplicationResponse": {
    "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
    "Id": "810c7aab86d42fb2b56c8c966example",
    "Name": "ExampleCorp",
    "tags": {
      "Stack": "Test"
    }
  }
}

```

- Per API i dettagli, vedere [CreateAppin AWS CLI Command Reference](#).

## create-sms-template

Il seguente esempio di codice mostra come utilizzare `create-sms-template`.

### AWS CLI

Crea un modello di messaggio per i messaggi inviati tramite il SMS canale

L'`create-sms-template` esempio seguente crea un modello di SMS messaggio.

```
aws pinpoint create-sms-template \  
  --template-name TestTemplate \  
  --sms-template-request file://myfile.json \  
  --region us-east-1
```

Contenuto di `myfile.json`.

```
{  
  "Body": "hello\n how are you?\n food is good",  
  "TemplateDescription": "Test SMS Template"  
}
```

Output:

```
{  
  "CreateTemplateMessageBody": {  
    "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:templates/  
TestTemplate/SMS",  
    "Message": "Created",  
    "RequestID": "8c36b17f-a0b0-400f-ac21-29e9b62a975d"  
  }  
}
```

Per ulteriori informazioni, consulta i [modelli di messaggi di Amazon Pinpoint nella Guida per l'utente di Amazon Pinpoint](#).

- Per API i dettagli, consulta AWS CLI Command [CreateSmsTemplateReference](#).

## delete-app

Il seguente esempio di codice mostra come utilizzare `delete-app`.

### AWS CLI

Eliminazione di un'applicazione

L'esempio `delete-app` seguente elimina un'applicazione (progetto).

```
aws pinpoint delete-app \  
  --application-id 810c7aab86d42fb2b56c8c966example
```

**Output:**

```
{
  "ApplicationResponse": {
    "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
    "Id": "810c7aab86d42fb2b56c8c966example",
    "Name": "ExampleCorp",
    "tags": {}
  }
}
```

- Per API i dettagli, vedere [DeleteApp](#) in AWS CLI Command Reference.

**get-apns-channel**

Il seguente esempio di codice mostra come utilizzare `get-apns-channel`.

**AWS CLI**

Per recuperare informazioni sullo stato e sulle impostazioni del APNs canale per un'applicazione

L'`get-apns-channel` esempio seguente recupera informazioni sullo stato e le impostazioni del APNs canale per un'applicazione.

```
aws pinpoint get-apns-channel \
  --application-id 9ab1068eb0a6461c86cce7f27ce0efd7 \
  --region us-east-1
```

**Output:**

```
{
  "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
  }
}

```

- Per API i dettagli, vedere [GetApsChannels](#) in AWS CLI Command Reference.

## get-app

Il seguente esempio di codice mostra come utilizzare `get-app`.

### AWS CLI

Per recuperare informazioni su un'applicazione (progetto)

L'`get-app` esempio seguente recupera informazioni su un'applicazione (progetto).

```

aws pinpoint get-app \
  --application-id 810c7aab86d42fb2b56c8c966example \
  --region us-east-1

```

Output:

```

{
  "ApplicationResponse": {
    "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
    "Id": "810c7aab86d42fb2b56c8c966example",
    "Name": "ExampleCorp",
    "tags": {
      "Year": "2019",
      "Stack": "Production"
    }
  }
}

```

- Per API i dettagli, vedere [GetApp](#) in AWS CLI Command Reference.

## get-apps

Il seguente esempio di codice mostra come utilizzare `get-apps`.

## AWS CLI

Per recuperare informazioni su tutte le tue applicazioni

L'get - appsempio seguente recupera informazioni su tutte le applicazioni (progetti).

```
aws pinpoint get-apps
```

Output:

```
{
  "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":
      "eyJJDcmVhdGlvbkRhdGUiOiIyMDE5LTA3LTE2VDE0jM40jUzLjkwM1oiLCJBY2NvdW50SWQiOiI1MTIzOTcxODM4Nz"
  }
}
```



```
}
}
```

La presenza del valore di NextToken risposta indica che c'è più output disponibile. Richiamate il comando e fornite quel valore come parametro NextToken di input.

- Per API i dettagli, vedere [GetApps](#) in AWS CLI Command Reference.

## get-campaign

Il seguente esempio di codice mostra come utilizzare `get-campaign`.

### AWS CLI

Per recuperare informazioni sullo stato, la configurazione e altre impostazioni di una campagna

L'`get-campaign` seguente recupera informazioni sullo stato, la configurazione e altre impostazioni di una campagna.

```
aws pinpoint get-campaign \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --campaign-id a1e63c6cc0eb43ed826ffcc3cc90b30d \
  --region us-east-1
```

Output:

```
{
  "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
}
}

```

- Per API i dettagli, vedere [GetCampaign](#) in AWS CLI Command Reference.

## get-campaigns

Il seguente esempio di codice mostra come utilizzare `get-campaigns`.

### AWS CLI

Per recuperare informazioni sullo stato, la configurazione e altre impostazioni per tutte le campagne associate a un'applicazione

L'esempio seguente recupera informazioni sullo stato, la configurazione e altre impostazioni per tutte le campagne associate a un'applicazione.

```
aws pinpoint get-campaigns \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1
```

## Output:

```
{
  "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  <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/
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=\n
\n  <head>\n    <meta http-equiv=\n
\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
  }
]
}
}

```

- Per API i dettagli, vedere [GetCampaigns](#) in AWS CLI Command Reference.

## get-channels

Il seguente esempio di codice mostra come utilizzare `get-channels`.

### AWS CLI

Per recuperare informazioni sulla cronologia e sullo stato di ogni canale per un'applicazione

L'`get-channel` esempio seguente recupera informazioni sulla cronologia e sullo stato di ogni canale per un'applicazione.

```

aws pinpoint get-channels \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1

```

Output:

```

{
  "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
    }
}
}
}
}
}

```

- Per API i dettagli, vedere [GetChannels](#) in AWS CLI Command Reference.

## get-email-channel

Il seguente esempio di codice mostra come utilizzare `get-email-channel`.

### AWS CLI

Per recuperare informazioni sullo stato e sulle impostazioni del canale di posta elettronica per un'applicazione

L'`get-email-channel` esempio seguente recupera lo stato e le impostazioni del canale Email per un'applicazione.

```

aws pinpoint get-email-channel \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1

```

Output:

```
{
  "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
  }
}
```

- Per API i dettagli, vedere [GetEmailChannel](#) in AWS CLI Command Reference.

## get-endpoint

Il seguente esempio di codice mostra come utilizzare `get-endpoint`.

### AWS CLI

Recupero di informazioni sulle impostazioni e gli attributi di un endpoint specifico per un'applicazione

L'esempio `get-endpoint` seguente recupera le informazioni sulle impostazioni e gli attributi di un endpoint specifico per un'applicazione.

```
aws pinpoint get-endpoint \
  --application-id 611e3e3cdd47474c9c1399a505665b91 \
  --endpoint-id testendpoint \
  --region us-east-1
```

Output:

```
{
  "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"
    }
  }
}

```

- Per API i dettagli, vedere [GetEndpoint](#) in AWS CLI Command Reference.

## get-gcm-channel

Il seguente esempio di codice mostra come utilizzare `get-gcm-channel`.

### AWS CLI

Per recuperare informazioni sullo stato e sulle impostazioni del GCM canale per un'applicazione

L'`get-gcm-channel` esempio seguente recupera informazioni sullo stato e le impostazioni del GCM canale per un'applicazione.

```

aws pinpoint get-gcm-channel \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \

```



```
--region us-east-1
```

Output:

```
{
  "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
  }
}
```

- Per API i dettagli, vedere [GetGcmChannel](#) in AWS CLI Command Reference.

## get-sms-channel

Il seguente esempio di codice mostra come utilizzare `get-sms-channel`.

### AWS CLI

Per recuperare informazioni sullo stato e sulle impostazioni del SMS canale per un'applicazione

L'esempio `get-sms-channel` seguente recupera lo stato e le impostazioni del canale di sms per un'applicazione.

```
aws pinpoint get-sms-channel \  
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \  
  --region us-east-1
```

Output:

```
{
  "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
  }
}

```

- Per API i dettagli, vedere [GetSmsChannel](#) in AWS CLI Command Reference.

## get-sms-template

Il seguente esempio di codice mostra come utilizzare `get-sms-template`.

### AWS CLI

Recupera il contenuto e le impostazioni di un modello di messaggio per i messaggi inviati tramite il canale SMS

L'esempio seguente recupera il contenuto e le impostazioni di un modello di SMS messaggio.

```

aws pinpoint get-sms-template \
  --template-name TestTemplate \
  --region us-east-1

```

Output:

```

{
  "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"
  }
}
```

Per ulteriori informazioni, consulta i [modelli di messaggi di Amazon Pinpoint nella Guida per l'utente di Amazon Pinpoint](#).

- Per API i dettagli, consulta AWS CLI Command [GetSmsTemplateReference](#).

## get-voice-channel

Il seguente esempio di codice mostra come utilizzare `get-voice-channel`.

### AWS CLI

Per recuperare informazioni sullo stato e sulle impostazioni del canale vocale di un'applicazione

L'`get-voice-channel` esempio seguente recupera lo stato e le impostazioni del canale vocale per un'applicazione.

```
aws pinpoint get-voice-channel \
  --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
  --region us-east-1
```

Output:

```
{
  "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
  }
}
```

- Per API i dettagli, vedere [GetVoiceChannel](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per recuperare un elenco di tag per una risorsa

L'`list-tags-for-resource` seguente recupera tutti i tag (nomi e valori chiave) associati alla risorsa specificata.

```
aws pinpoint list-tags-for-resource \
  --resource-arn arn:aws:mobiletargeting:us-west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example
```

Output:

```
{
  "TagsModel": {
    "tags": {
      "Year": "2019",
      "Stack": "Production"
    }
  }
}
```

Per ulteriori informazioni, consulta 'Tagging Amazon Pinpoint Resources <https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>' nella Amazon Pinpoint Developer Guide.

- Per i API dettagli, [ListTagsForResource](#) consulta AWS CLI Command Reference.

## phone-number-validate

Il seguente esempio di codice mostra come utilizzare `phone-number-validate`.

### AWS CLI

Recupera informazioni su un numero di telefono

Quanto segue `phone-number-validate` recupera le informazioni su un numero di telefono.

```
aws pinpoint phone-number-validate \  
  --number-validate-request PhoneNumber="+12065550142" \  
  --region us-east-1
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta il [SMS canale Amazon Pinpoint](#) nella Guida per l'utente di Amazon Pinpoint.

- Per API i dettagli, consulta AWS CLI Command [PhoneNumberValidate](#) Reference.

## send-messages

Il seguente esempio di codice mostra come utilizzare `send-messages`.

### AWS CLI

Per inviare SMS un messaggio utilizzando l'endpoint di un'applicazione

L'esempio `send-messages` seguente invia un messaggio diretto per un'applicazione con un endpoint.

```
aws pinpoint send-messages \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --message-request file://myfile.json \  
  --region us-east-1
```

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

Contenuto di `myfile.json`.

```
{
  "MessageConfiguration": {
    "SMSMessage": {
      "Body": "hello, how are you?"
    }
  },
  "Endpoints": {
    "testendpoint": {}
  }
}
```

Output:

```
{
  "MessageResponse": {
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
    "EndpointResult": {
      "testendpoint": {
        "Address": "+12345678900",
        "DeliveryStatus": "SUCCESSFUL",
        "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb31q6g0",
        "StatusCode": 200,
        "StatusMessage": "MessageId:
itnuqhai5alf1n6ahv3udc05n7hhddr6gb31q6g0"
      }
    },
    "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"
  }
}
```

Per ulteriori informazioni, consulta il [SMS canale Amazon Pinpoint](#) nella Guida per l'utente di Amazon Pinpoint.

- Per API i dettagli, consulta AWS CLI Command [SendMessages](#) Reference.

## send-users-messages

Il seguente esempio di codice mostra come utilizzare `send-users-messages`.

## AWS CLI

Per inviare un SMS messaggio a un utente di un'applicazione

L'`send-users-messages` seguente invia un messaggio diretto per un utente di un'applicazione.

```
aws pinpoint send-users-messages \  
  --application-id 611e3e3cdd47474c9c1399a505665b91 \  
  --send-users-message-request file://myfile.json \  
  --region us-west-2
```

Contenuto di `myfile.json`.

```
{  
  "MessageConfiguration": {  
    "SMSMessage": {  
      "Body": "hello, how are you?"  
    }  
  },  
  "Users": {  
    "testuser": {}  
  }  
}
```

Output:

```
{  
  "SendUsersMessageResponse": {  
    "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",  
    "RequestId": "e0b12cf5-2359-11e9-bb0b-d5fb91876b25",  
    "Result": {  
      "testuser": {  
        "testuserendpoint": {  
          "DeliveryStatus": "SUCCESSFUL",  
          "MessageId": "7qu4hk5bqhda3i7i2n4pjf98qcu8b7p45ifsmo0",  
          "StatusCode": 200,  
          "StatusMessage": "MessageId:  
7qu4hk5bqhda3i7i2n4pjf98qcu8b7p45ifsmo0",  
          "Address": "+12345678900"  
        }  
      }  
    }  
  }  
}
```

```
}  
}
```

Per ulteriori informazioni, consulta il [SMS canale Amazon Pinpoint](#) nella Guida per l'utente di Amazon Pinpoint.

- Per API i dettagli, consulta AWS CLI Command [SendUsersMessages](#) Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere tag a una risorsa

L'esempio seguente aggiunge due tag (nomi e valori chiave) a una risorsa.

```
aws pinpoint list-tags-for-resource \  
  --resource-arn arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \  
  --tags-model tags={Stack=Production,Year=2019}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta 'Tagging Amazon Pinpoint Resources [https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging <-resources.html>](https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html)' nella Amazon Pinpoint Developer Guide.

- Per i API dettagli, [TagResource](#) consulta AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Esempio 1: rimuovere un tag da una risorsa

L'`untag-resource` esempio seguente rimuove il tag specificato (nome e valore della chiave) da una risorsa.

```
aws pinpoint untag-resource \  

```



```
--resource-arn arn:aws:mobiletargeting:us-west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \  
--tag-keys Year
```

Questo comando non produce alcun output.

Esempio 2: rimuovere più tag da una risorsa

L'`untag-resource` seguente rimuove i tag specificati (nomi e valori delle chiavi) da una risorsa.

```
aws pinpoint untag-resource \  
--resource-arn arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \  
--tag-keys Year Stack
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta 'Tagging Amazon Pinpoint Resources <https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>' nella Amazon Pinpoint Developer Guide.

- Per i API dettagli, [UntagResource](#) consulta AWS CLI Command Reference.

## update-sms-channel

Il seguente esempio di codice mostra come utilizzare `update-sms-channel`.

AWS CLI

Per abilitare il SMS canale o aggiornare lo stato e le impostazioni del SMS canale per un'applicazione.

L'`update-sms-channel` seguente abilita il SMS canale per un SMS canale per un'applicazione.

```
aws pinpoint update-sms-channel \  
--application-id 611e3e3cdd47474c9c1399a505665b91 \  
--sms-channel-request Enabled=true \  
--region us-west-2
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta il [SMS canale Amazon Pinpoint](#) nella Guida per l'utente di Amazon Pinpoint.

- Per API i dettagli, consulta AWS CLI Command [UpdateSmsChannelReference](#).

## Esempi di utilizzo di Amazon Polly AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Polly. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **delete-lexicon**

Il seguente esempio di codice mostra come utilizzare `delete-lexicon`.

## AWS CLI

Per eliminare un lessico

L'`delete-lexicon` seguente elimina il lessico specificato.

```
aws polly delete-lexicon \  
  --name w3c
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Utilizzo dell' DeleteLexicon operazione](#) nella Amazon Polly Developer Guide.

- Per API i dettagli, consulta [DeleteLexicon AWS CLI Command Reference](#).

## get-lexicon

Il seguente esempio di codice mostra come utilizzare `get-lexicon`.

## AWS CLI

Per recuperare il contenuto di un lessico

L'`get-lexicon` seguente recupera il contenuto del lessico di pronuncia specificato.

```
aws polly get-lexicon \  
  --name w3c
```

Output:

```
{  
  "Lexicon": {  
    "Content": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<lexicon version=  
\"1.0\" \n      xmlns=      \"http://www.w3.org/2005/01/pronunciation-lexicon  
\" \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
}
```

Per ulteriori informazioni, consulta [Utilizzo dell' GetLexicon operazione](#) nella Amazon Polly Developer Guide.

- Per API i dettagli, consulta [GetLexicon AWS CLI Command Reference](#).

## get-speech-synthesis-task

Il seguente esempio di codice mostra come utilizzare `get-speech-synthesis-task`.

### AWS CLI

Per ottenere informazioni su un'operazione di sintesi vocale

L'`get-speech-synthesis-task` esempio seguente recupera informazioni sull'attività di sintesi vocale specificata.

```
aws polly get-speech-synthesis-task \
  --task-id 70b61c0f-57ce-4715-a247-cae8729dcce9
```

Output:

```
{
  "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"
  }
}
```

```
}  
}
```

Per ulteriori informazioni, consulta [Creazione di file audio lunghi](#) nella Amazon Polly Developer Guide.

- Per API i dettagli, consulta [GetSpeechSynthesisTask AWS CLI Command Reference](#).

## list-lexicons

Il seguente esempio di codice mostra come utilizzare `list-lexicons`.

### AWS CLI

Per elencare i tuoi lessici

L'`list-lexicons` esempio seguente elenca i tuoi lessici di pronuncia.

```
aws polly list-lexicons
```

Output:

```
{  
  "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  
      }  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Utilizzo dell' ListLexicons operazione](#) nella Amazon Polly Developer Guide.

- Per API i dettagli, consulta [ListLexicons AWS CLI Command Reference](#).

## list-speech-synthesis-tasks

Il seguente esempio di codice mostra come utilizzare `list-speech-synthesis-tasks`.

### AWS CLI

Per elencare le attività di sintesi vocale

L'`list-speech-synthesis-tasks` seguente elenca le attività di sintesi vocale.

```
aws polly list-speech-synthesis-tasks
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Creazione di file audio lunghi](#) nella Amazon Polly Developer Guide.

- Per API i dettagli, consulta [ListSpeechSynthesisTasks AWS CLI Command Reference](#).

## put-lexicon

Il seguente esempio di codice mostra come utilizzare `put-lexicon`.

### AWS CLI

Per memorizzare un lessico

L'`put-lexicon` esempio seguente memorizza il lessico di pronuncia specificato. Il `example.pls` file specifica un lessico conforme al W3CPLS.

```
aws polly put-lexicon \  
  --name w3c \  
  --content file://example.pls
```

Contenuto di `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>  
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Utilizzo dell' PutLexicon operazione](#) nella Amazon Polly Developer Guide.

- Per API i dettagli, consulta [PutLexicon AWS CLI Command Reference](#).

## start-speech-synthesis-task

Il seguente esempio di codice mostra come utilizzare `start-speech-synthesis-task`.

AWS CLI

Per sintetizzare il testo

L'`start-speech-synthesis-task` esempio seguente sintetizza il testo `text_file.txt` e memorizza il MP3 file risultante nel bucket specificato.

```
aws polly start-speech-synthesis-task \  
  --output-format mp3 \  
  --output-s3-bucket-name my-s3-bucket \  
  --text file://text_file.txt \  
  --voice-id Joanna
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Creazione di file audio lunghi](#) nella Amazon Polly Developer Guide.

- Per API i dettagli, consulta [StartSpeechSynthesisTask AWS CLI Command Reference](#).

## Listino prezzi AWS esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Listino prezzi AWS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)



## Azioni

### **describe-services**

Il seguente esempio di codice mostra come utilizzare `describe-services`.

#### AWS CLI

Per recuperare i metadati del servizio

Questo esempio recupera i metadati per il codice di EC2 servizio Amazon.

Comando:

```
aws pricing describe-services --service-code AmazonEC2 --format-version aws_v1 --  
max-items 1
```

Output:

```
{  
  "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"  
}
```

- Per API i dettagli, consulta AWS CLI Command [DescribeServicesReference](#).

## get-attribute-values

Il seguente esempio di codice mostra come utilizzare `get-attribute-values`.

### AWS CLI

Per recuperare un elenco di valori degli attributi

L'`get-attribute-values` esempio seguente recupera un elenco di valori disponibili per l'attributo specificato.

```
aws pricing get-attribute-values \  
  --service-code AmazonEC2 \  
  --attribute-name volumeType \  
  --max-items 2
```

Output:

```
{  
  "NextToken": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ==",  
  "AttributeValues": [  
    {  
      "Value": "Cold HDD"  
    },  
    {  
      "Value": "General Purpose"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [GetAttributeValues](#) in AWS CLI Command Reference.

## get-products

Il seguente esempio di codice mostra come utilizzare `get-products`.

## AWS CLI

Per recuperare un elenco di prodotti

Questo esempio recupera un elenco di prodotti che corrispondono ai criteri indicati.

Comando:

```
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"
  }
]
```

Output:

```
{
  "FormatVersion": "aws_v1",
  "NextToken": "WGDY7ko8fQXdlaUZVdasFQ==:RVsagyIFn770XQ0zdUIc09BY6ucBG9itXAZGZF/zioUz0sUKh6PCcPwa0yPZRiMePb986TeoKYB9l55fw/CyoMq5ymnGmT1Vj39Tljbba1hcqnVfTmPIilx8Uy5bdDaBYy/e/20fw9Edzsykbs8LTBUbNbiDQ+BBds5yeI9AQkUepruKk3aEahFPxJ55kx/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\": \"3MKHN58N7RDDVVGKJ\" }, \"serviceCode\": \"AmazonEC2\", \"terms\": { \"OnDemand\": { \"3MKHN58N7RDDVVGKJ.JRTCKXETXF\": { \"priceDimensions\": { \"3MKHN58N7RDDVVGKJ.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\"}
  ]
}
```

- Per API i dettagli, vedere [GetProducts](#) in AWS CLI Command Reference.

## AWS Private CA esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Private CA.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **create-certificate-authority-audit-report**

Il seguente esempio di codice mostra come utilizzare `create-certificate-authority-audit-report`.

#### AWS CLI

Per creare un rapporto di controllo dell'autorità di certificazione

Il `create-certificate-authority-audit-report` comando seguente crea un rapporto di controllo per la CA privata identificata da ARN.

```
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
```

- Per API i dettagli, vedere [CreateCertificateAuthorityAuditReport](#) in AWS CLI Command Reference.

## create-certificate-authority

Il seguente esempio di codice mostra come utilizzare `create-certificate-authority`.

### AWS CLI

Per creare un'autorità di certificazione privata

Il `create-certificate-authority` comando seguente crea un'autorità di certificazione privata nel tuo AWS account.

```
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
```

- Per API i dettagli, vedere [CreateCertificateAuthority](#) in AWS CLI Command Reference.

## delete-certificate-authority

Il seguente esempio di codice mostra come utilizzare `delete-certificate-authority`.

### AWS CLI

Per eliminare un'autorità di certificazione privata

Il `delete-certificate-authority` comando seguente elimina l'autorità di certificazione identificata da. ARN

```
aws acm-pca delete-certificate-authority --certificate-  
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-  
authority/12345678-1234-1234-1234-123456789012
```

- Per API i dettagli, vedere [DeleteCertificateAuthority](#) in AWS CLI Command Reference.

## describe-certificate-authority-audit-report

Il seguente esempio di codice mostra come utilizzare `describe-certificate-authority-audit-report`.

### AWS CLI

Per descrivere un rapporto di controllo per un'autorità di certificazione

Il `describe-certificate-authority-audit-report` comando seguente elenca le informazioni sul rapporto di controllo specificato per la CA identificata da 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
```

- Per API i dettagli, vedere [DescribeCertificateAuthorityAuditReport](#) in AWS CLI Command Reference.

## describe-certificate-authority

Il seguente esempio di codice mostra come utilizzare `describe-certificate-authority`.

### AWS CLI

Per descrivere un'autorità di certificazione privata

Il `describe-certificate-authority` comando seguente elenca le informazioni sulla CA privata identificata da 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
```

- Per API i dettagli, vedere [DescribeCertificateAuthority](#) in AWS CLI Command Reference.

## get-certificate-authority-certificate

Il seguente esempio di codice mostra come utilizzare `get-certificate-authority-certificate`.

## AWS CLI

Per recuperare un certificato di autorità di certificazione (CA)

Il `get-certificate-authority-certificate` comando seguente recupera il certificato e la catena di certificati per la CA privata specificata da. ARN

```
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
```

- Per API i dettagli, vedere [GetCertificateAuthorityCertificate](#) in AWS CLI Command Reference.

## `get-certificate-authority-csr`

Il seguente esempio di codice mostra come utilizzare `get-certificate-authority-csr`.

## AWS CLI

Per recuperare la richiesta di firma del certificato per un'autorità di certificazione

Il `get-certificate-authority-csr` comando seguente recupera CSR la CA privata specificata da. ARN

```
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
```

- Per API i dettagli, vedere [GetCertificateAuthorityCsr](#) in AWS CLI Command Reference.

## `get-certificate`

Il seguente esempio di codice mostra come utilizzare `get-certificate`.

## AWS CLI

Per recuperare un certificato emesso

L'`get-certificate` esempio seguente recupera un certificato dalla CA privata specificata.

```
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

```

Output:

```

-----BEGIN CERTIFICATE-----
MIIEDzCCAvegAwIBAgIRAJuJ8f6ZVYL7gG/rS3qvrZMwDQYJKoZIhvcNAQELBQAw
cTElMAkGA1UEBhMVCVVMxEzARBgNVBAGMCl dhc2hpbmd0b24xEDA0BgNVBACMB1Nl
...certificate body truncated for brevity...
tKCSglgZZrd4FdLw1EkGm+UVXnodwMtJEQyy3oTfZjURPIyyaqskTu/KSS7YDjk0
KQNY73D6Ltmd0EbAyyq10XiDxqY41lvKHJ1eZrPaBmYNABxU=
-----END CERTIFICATE----- -----BEGIN CERTIFICATE-----
MIIDrzCCApegAwIBAgIRA0skdzLvcj1eShkoyEE693AwDQYJKoZIhvcNAQELBQAw
cTElMAkGA1UEBhMVCVVMxEzARBgNVBAGMCl dhc2hpbmd0b24xEDA0BgNVBACMB1Nl
...certificate body truncated for brevity...
kdRGB6P2hpxstDOUIwAoCbhoaWwfA4ybJzmf+j0QhAziNlRdKQRR8nODWpKt7H9w
dJ5nxsTk/fniJz86Ddtp6n8s82wYdkN3cVffeK72A9aTCOU=
-----END CERTIFICATE-----

```

La prima parte dell'output è il certificato stesso. La seconda parte è la catena di certificati che si collega al certificato CA principale. Si noti che quando si utilizza l' `--output text` opzione, viene inserito un TAB carattere tra le due parti del certificato (questa è la causa del testo rientrato). Se intendete prendere questo risultato e analizzare i certificati con altri strumenti, potreste dover rimuovere il TAB carattere in modo che venga elaborato correttamente.

- Per API i dettagli, consultate [GetCertificate AWS CLI](#) Command Reference.

## **import-certificate-authority-certificate**

Il seguente esempio di codice mostra come utilizzare `import-certificate-authority-certificate`.

### AWS CLI

Per importare il certificato dell'autorità di certificazione in ACM PCA

Il `import-certificate-authority-certificate` comando seguente importa il certificato CA privato firmato per la CA specificata da ARN into 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
```

- Per API i dettagli, vedere [ImportCertificateAuthorityCertificate](#) in AWS CLI Command Reference.

## issue-certificate

Il seguente esempio di codice mostra come utilizzare `issue-certificate`.

### AWS CLI

Per emettere un certificato privato

Il `issue-certificate` comando seguente utilizza la CA privata specificata da ARN per emettere un certificato privato.

```
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
```

- Per API i dettagli, vedere [IssueCertificate](#) in AWS CLI Command Reference.

## list-certificate-authorities

Il seguente esempio di codice mostra come utilizzare `list-certificate-authorities`.

### AWS CLI

Per elencare le autorità di certificazione private

Il `list-certificate-authorities` comando seguente elenca le informazioni su tutti i dati privati CAs presenti nel tuo account.

```
aws acm-pca list-certificate-authorities --max-results 10
```

- Per API i dettagli, vedere [ListCertificateAuthorities](#) in AWS CLI Command Reference.

## list-tags

Il seguente esempio di codice mostra come utilizzare `list-tags`.

### AWS CLI

Per elencare i tag relativi all'autorità di certificazione

Il `list-tags` comando seguente elenca i tag associati alla CA privata specificata da ARN.

```
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
```

- Per API i dettagli, vedere [ListTags](#) in AWS CLI Command Reference.

## revoke-certificate

Il seguente esempio di codice mostra come utilizzare `revoke-certificate`.

### AWS CLI

Per revocare un certificato privato

Il `revoke-certificate` comando seguente revoca un certificato privato dalla CA identificata da ARN

```
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"
```

- Per API i dettagli, vedere [RevokeCertificate](#) in AWS CLI Command Reference.

## tag-certificate-authority

Il seguente esempio di codice mostra come utilizzare `tag-certificate-authority`.

### AWS CLI

Per allegare tag a un'autorità di certificazione privata

Il `tag-certificate-authority` comando seguente allega uno o più tag alla tua CA privata.

```
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
```

- Per API i dettagli, vedere [TagCertificateAuthority](#) in AWS CLI Command Reference.

## `untag-certificate-authority`

Il seguente esempio di codice mostra come utilizzare `untag-certificate-authority`.

### AWS CLI

Per rimuovere uno o più tag dall'autorità di certificazione privata

Il `untag-certificate-authority` comando seguente rimuove i tag dalla CA privata identificata da ARN.

```
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
```

- Per API i dettagli, vedere [UntagCertificateAuthority](#) in AWS CLI Command Reference.

## `update-certificate-authority`

Il seguente esempio di codice mostra come utilizzare `update-certificate-authority`.

### AWS CLI

Per aggiornare la configurazione dell'autorità di certificazione privata

Il `update-certificate-authority` comando seguente aggiorna lo stato e la configurazione della CA privata identificata da ARN.

```
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"
```

- Per API i dettagli, vedere [UpdateCertificateAuthority](#) in AWS CLI Command Reference.

## AWS Proton esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Proton.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

### Azioni

#### **cancel-service-instance-deployment**

Il seguente esempio di codice mostra come utilizzare `cancel-service-instance-deployment`.

AWS CLI

Per annullare la distribuzione di un'istanza di servizio

L'`cancel-service-instance-deployment` seguente annulla la distribuzione di un'istanza di servizio.

```
aws proton cancel-service-instance-deployment \  
  --service-instance-name "instance-one" \  
  --service-name "simple-svc"
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, vedere [Aggiornare un'istanza di servizio](#) nella Guida per l'amministratore di AWS Proton o [Aggiornare un'istanza di servizio](#) nella Guida per l'utente di AWS Proton.

- Per API i dettagli, vedere [CancelServiceInstanceDeployment](#) in AWS CLI Command Reference.

## cancel-service-pipeline-deployment

Il seguente esempio di codice mostra come utilizzare `cancel-service-pipeline-deployment`.

### AWS CLI

Per annullare la distribuzione di una pipeline di servizi

L'`cancel-service-pipeline-deployment` seguente annulla la distribuzione di una pipeline di servizi.

```
aws proton cancel-service-pipeline-deployment \
  --service-name "simple-svc"
```

Output:

```
{
```

```

"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"
}
}

```

Per ulteriori informazioni, vedere [Aggiornare una pipeline di servizi](#) nella The AWS Proton Administrator Guide o [Aggiornare una pipeline di servizi](#) nella The AWS Proton User Guide.

- Per i API dettagli, vedere [CancelServicePipelineDeployment](#) in Command Reference.AWS CLI

## create-service

Il seguente esempio di codice mostra come utilizzare `create-service`.

### AWS CLI

Per creare un servizio

L'`create-service`esempio seguente crea un servizio con una pipeline di servizi.

```

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

```

Contenuto di `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"

```

### Output:

```

{
  "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:123456789012:connection/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "repositoryId": "myorg/myapp",
    "status": "CREATE_IN_PROGRESS",
    "templateName": "fargate-service"
  }
}

```

Per ulteriori informazioni, vedere [Creare un servizio](#) nella Guida per l'amministratore di AWS Proton e [Creare un servizio](#) nella Guida per l'utente di AWS Proton.

- Per API i dettagli, vedere [CreateService](#) in AWS CLI Command Reference.

## delete-service

Il seguente esempio di codice mostra come utilizzare `delete-service`.

### AWS CLI

Per eliminare un servizio

L'esempio seguente elimina un servizio.

```

aws proton delete-service \
  --name "simple-svc"

```



**Output:**

```
{
  "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"
  }
}
```

Per ulteriori informazioni, vedere [Eliminare un servizio](#) nella The AWS Proton Administrator Guide.

- Per API i dettagli, vedere [DeleteService](#) in AWS CLI Command Reference.

**get-service-instance**

Il seguente esempio di codice mostra come utilizzare `get-service-instance`.

**AWS CLI**

Per ottenere i dettagli dell'istanza del servizio

L'`get-service-instance` esempio seguente ottiene dati di dettaglio per un'istanza di servizio.

```
aws proton get-service-instance \
  --name "instance-one" \
  --service-name "simple-svc"
```

**Output:**

```
{
  "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"
  }
}

```

Per ulteriori informazioni, vedere [Visualizza i dati di servizio](#) nella The AWS Proton Administrator Guide o [Visualizza i dati di servizio](#) nella The AWS Proton User Guide.

- Per API i dettagli, vedere [GetServiceInstance](#) in AWS CLI Command Reference.

## get-service

Il seguente esempio di codice mostra come utilizzare `get-service`.

### AWS CLI

Per ottenere i dettagli del servizio

L'`get-service` esempio seguente ottiene dati di dettaglio per un servizio.

```

aws proton get-service \
  --name "simple-svc"

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, vedere [Visualizza i dati di servizio](#) nella The AWS Proton Administrator Guide o [Visualizza i dati di servizio](#) nella The AWS Proton User Guide.

- Per API i dettagli, vedere [GetService](#) in AWS CLI Command Reference.

## list-service-instances

Il seguente esempio di codice mostra come utilizzare `list-service-instances`.

### AWS CLI

Esempio 1: per elencare tutte le istanze del servizio

L'`list-service-instances` seguente elenca le istanze del servizio.

```
aws proton list-service-instances
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Visualizza i dati dell'istanza del servizio](#) nella Guida per l'amministratore di AWS Proton o [Visualizza i dati dell'istanza del servizio](#) nella Guida per l'utente di AWS Proton.

Esempio 2: per elencare l'istanza di servizio specificata

L'`get-service-instance` seguente ottiene un'istanza di servizio.

```
aws proton get-service-instance \
  --name "instance-one" \
  --service-name "simple-svc"
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, vedere [Visualizza i dati dell'istanza del servizio](#) nella Guida per l'amministratore di AWS Proton o [Visualizza i dati dell'istanza del servizio](#) nella Guida per l'utente di AWS Proton.

- Per API i dettagli, vedere [ListServiceInstances](#) in AWS CLI Command Reference.

## update-service-instance

Il seguente esempio di codice mostra come utilizzare `update-service-instance`.

### AWS CLI

Per aggiornare un'istanza del servizio a una nuova versione secondaria

L'update-service-instanceesempio seguente aggiorna un'istanza del servizio con una nuova versione secondaria del relativo modello di servizio che aggiunge una nuova istanza denominata `my-other-instance` con un nuovo input richiesto.

```

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"
```

Contenuto di `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"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Aggiornare un'istanza di servizio](#) nella Guida per l'amministratore di AWS Proton o [Aggiornare un'istanza di servizio](#) nella Guida per l'utente di AWS Proton.

- Per API i dettagli, vedere [UpdateServiceInstance](#) in AWS CLI Command Reference.

## update-service-pipeline

Il seguente esempio di codice mostra come utilizzare `update-service-pipeline`.

### AWS CLI

Per aggiornare una pipeline di servizi

L'esempio seguente aggiorna una pipeline di servizi con una nuova versione secondaria del relativo modello di servizio.

```
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"
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, vedere [Aggiornare una pipeline di servizi](#) nella The AWS Proton Administrator Guide o [Aggiornare una pipeline di servizi](#) nella The AWS Proton User Guide.

- Per i API dettagli, vedere [UpdateServicePipeline](#) in Command Reference.AWS CLI

## update-service

Il seguente esempio di codice mostra come utilizzare `update-service`.

### AWS CLI

Per aggiornare un servizio

L'`update-service` esempio seguente modifica la descrizione di un servizio.

```
aws proton update-service \  
  --name "MySimpleService" \  
  --description "Edit by updating description"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Modificare un servizio](#) nella The AWS Proton Administrator Guide o [Modificare un servizio](#) nella The AWS Proton User Guide.

- Per API i dettagli, vedere [UpdateService](#) in AWS CLI Command Reference.



## QLDB esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with QLDB.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **cancel-journal-kinesis-stream**

Il seguente esempio di codice mostra come utilizzare `cancel-journal-kinesis-stream`.

AWS CLI

Per annullare lo streaming di un diario

L'`cancel-journal-kinesis-stream` esempio seguente annulla lo stream di journal specificato da un libro mastro.

```
aws qldb cancel-journal-kinesis-stream \  
  --ledger-name myExampleLedger \  
  --stream-id 7ISckqwe4y25YyHLzYUFAf
```

Output:

```
{  
  "StreamId": "7ISckqwe4y25YyHLzYUFAf"  
}
```

Per ulteriori informazioni, consulta [Streaming journal data from Amazon QLDB nell'Amazon QLDB Developer Guide](#).

- Per API i dettagli, consulta [CancelJournalKinesisStream AWS CLI Command Reference](#).

## create-ledger

Il seguente esempio di codice mostra come utilizzare `create-ledger`.

### AWS CLI

Esempio 1: creare un libro mastro con proprietà predefinite

L'`create-ledger` esempio seguente crea un libro mastro con il nome `myExampleLedger` e la modalità di autorizzazione. `STANDARD` I parametri opzionali per la protezione dall'eliminazione e la AWS KMS chiave non sono specificati, quindi per impostazione predefinita sono rispettivamente una chiave proprietaria `true` e una KMS chiave AWS proprietaria.

```
aws qlldb create-ledger \  
  --name myExampleLedger \  
  --permissions-mode STANDARD
```

Output:

```
{  
  "State": "CREATING",  
  "Arn": "arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger",  
  "DeletionProtection": true,  
  "CreationDateTime": 1568839243.951,  
  "Name": "myExampleLedger",  
  "PermissionsMode": "STANDARD"  
}
```

Esempio 2: creare un registro con la protezione da eliminazione disattivata, una KMS chiave gestita dal cliente e tag specificati

L'`create-ledger` esempio seguente crea un libro mastro con il nome `myExampleLedger2` e la modalità di autorizzazione. `STANDARD` La funzionalità di protezione dall'eliminazione è disattivata, la KMS chiave gestita dal cliente specificata viene utilizzata per la crittografia a riposo e i tag specificati vengono allegati alla risorsa.

```
aws qlldb create-ledger \  
  --name myExampleLedger2 \  
  --tags key=value
```

```
--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
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Basic Operations for Amazon QLDB Ledgers](#) nella Amazon QLDB Developer Guide.

- Per API i dettagli, consulta [CreateLedger AWS CLI Command Reference](#).

## delete-ledger

Il seguente esempio di codice mostra come utilizzare `delete-ledger`.

AWS CLI

Per eliminare un libro mastro

L'`delete-ledger` esempio seguente elimina il libro mastro specificato.

```
aws qldb delete-ledger \  
  --name myExampleLedger
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Basic Operations for Amazon QLDB Ledgers](#) nella Amazon QLDB Developer Guide.

- Per API i dettagli, consulta [DeleteLedger AWS CLI Command Reference](#).

## describe-journal-kinesis-stream

Il seguente esempio di codice mostra come utilizzare `describe-journal-kinesis-stream`.

### AWS CLI

Per descrivere lo stream di un diario

L'esempio seguente visualizza i dettagli per il flusso di journal specificato da un libro mastro.

```
aws qlldb describe-journal-kinesis-stream \  
  --ledger-name myExampleLedger \  
  --stream-id 7ISCKqwe4y25YyHLzYUFaf
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Streaming journal data from Amazon QLDB nell'Amazon QLDB Developer Guide](#).

- Per API i dettagli, consulta [DescribeJournalKinesisStream AWS CLI Command Reference](#).

## describe-journal-s3-export

Il seguente esempio di codice mostra come utilizzare `describe-journal-s3-export`.

### AWS CLI

Per descrivere un processo di esportazione di un diario

L'esempio seguente visualizza i dettagli del processo di esportazione specificato da un libro mastro.

```
aws qlldb describe-journal-s3-export \  
  --name myExampleLedger \  
  --export-id ADR20NPKN5LINYGb4dp7yZ
```

### Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Exporting Your Journal in Amazon QLDB nella Amazon QLDB Developer Guide](#).

- Per API i dettagli, consulta [DescribeJournalS3Export](#) in AWS CLI Command Reference.

## describe-ledger

Il seguente esempio di codice mostra come usare `describe-ledger`

### AWS CLI

Per descrivere un libro mastro

L'esempio seguente visualizza i dettagli del libro contabile specificato.

```
aws qldb describe-ledger \  
  --name myExampleLedger
```

Output:

```
{  
  "CreationDateTime": 1568839243.951,  
  "Arn": "arn:aws:qldb: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"  
  }  
}
```

Per ulteriori informazioni, consulta [Basic Operations for Amazon QLDB Ledgers](#) nella Amazon QLDB Developer Guide.

- Per API i dettagli, consulta [DescribeLedger AWS CLI Command Reference](#).

## export-journal-to-s3

Il seguente esempio di codice mostra come utilizzare `export-journal-to-s3`.

### AWS CLI

Per esportare blocchi di journal in S3

L'export-journal-to-s3 esempio seguente crea un processo di esportazione per i blocchi di diario entro un intervallo di data e ora specificato da un registro con il nome `myExampleLedger`. Il processo di esportazione scrive i blocchi in un bucket Amazon S3 specificato.

```
aws qldb 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
```

Contenuto di `my-s3-export-config.json`.

```
{  
  "Bucket": "awsExampleBucket",  
  "Prefix": "ledgerexport1/",  
  "EncryptionConfiguration": {  
    "ObjectEncryptionType": "SSE_S3"  
  }  
}
```

Output:

```
{  
  "ExportId": "ADR20NPKN5LINYGb4dp7yZ"  
}
```

Per ulteriori informazioni, consulta [Exporting Your Journal in Amazon QLDB nella Amazon QLDB Developer Guide](#).

- Per API i dettagli, consulta [ExportJournalToS3](#) in AWS CLI Command Reference.

## get-block

Il seguente esempio di codice mostra come usare `get-block`.

### AWS CLI

Esempio 1: Per ottenere un blocco di diario e una bozza per la verifica utilizzando i file di input

L'`get-block` esempio seguente richiede un oggetto di dati a blocchi e una bozza dal registro specificato. La richiesta riguarda un indirizzo digest tip e un indirizzo di blocco specificati.

```
aws qlldb get-block \
  --name vehicle-registration \
  --block-address file://myblockaddress.json \
  --digest-tip-address file://mydigesttipaddress.json
```

Contenuto di myblockaddress.json.

```
{
  "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iw1\",sequenceNo:100}"
}
```

Contenuto di mydigesttipaddress.json.

```
{
  "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iw1\",sequenceNo:123}"
}
```

Output:

```
{
  "Block": {
    "IonText": "{blockAddress:{strandId:\"KmA3ZZca7vAIiJAK9S5Iw1
  \",sequenceNo:100},transactionId:\"FnQeJBAicTX0Ah32ZnVtSX
  \",blockTimestamp:2019-09-16T19:37:05.360Z,blockHash:
  {{NoChM92yKRuJAb/jeLd1VnYn4DHiWIf071ACfic9uHc=}},entriesHash:
  {{105L0siKV14SDbuaYnH7uwXzUvqzIwUiRLXGbTyj/nY=}},previousBlockHash:
  {{7kewBXhpdBc1cZKxhVmpoMhpUG0JtWQD0iY2LPfZkYA=}},entriesHashList:
  [{{eRSwnmAM7WWANWDd5iG0yK+T4tDXyzUq6HZ/0fgLHos=}},{{mHVex/
  yjHAWjFPpwhBuH2GKXmKjK2FBa9faquUVNtg=}},
  {{y5cCB7p0AIUfsVQ1j0TqtE97b4b4oo1R0vnYyE5wWM=}},{{TvTXygML1bMe6NvEZtGkX
  +KR+W/EJl4qD1mmV77KZQg=}}}],transactionInfo:{statements:[{statement:
  \"FROM VehicleRegistration AS r \\nWHERE r.VIN = '1N4AL11D75C109151'\\n
  \\nINSERT INTO r.Owners.SecondaryOwners\\n  VALUE { 'PersonId' :
  'CMVdR77XP8zAg1mmFDGTvt' }\\n\",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/yjHAWjFPpwhBuH2GKXmKjK2FBa9faquUVNtg=}},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\": \"[{{l3+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},
{{48CXG3ehPqsxCYd34EEa8Fso00RpWwA08010RJKf3Do=}},{{9UnwnKSQT0i3ge1JMVa
+tMIqCEDa0PTkwxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFctLufgPM6qXHyTNeCb1sCwcDaI=}},
{{Irb5fNhbRiNEQ1VPhzlnGT/ZQPadSmgfdtMYcwkN0xoI=}},{{+3CwpYG/ytf/
vq9GidpzSx6JJiLXt1hMQWnNq0y3jfY=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT
+qE=}}}]\"
  }
}
```

Per ulteriori informazioni, consulta la sezione [Verifica dei dati in Amazon QLDB](#) nella Amazon QLDB Developer Guide.

Esempio 2: Per ottenere un blocco di diario e una bozza per la verifica, utilizza una sintassi abbreviata

L'get-block esempio seguente richiede un oggetto di dati a blocchi e una bozza dal registro specificato utilizzando una sintassi abbreviata. La richiesta riguarda un indirizzo digest tip e un indirizzo di blocco specificati.

```
aws qldb get-block \
  --name vehicle-registration \
  --block-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5IwI\",sequenceNo:100}"'
  \
  --digest-tip-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5IwI\",sequenceNo:123}"'
```

Output:

```
{
  \"Block\": {
    \"IonText\": \"{blockAddress:{strandId:\"KmA3ZZca7vAIiJAK9S5IwI
\\\",sequenceNo:100},transactionId:\"FnQeJBAicTX0Ah32ZnVtSX
\\\",blockTimestamp:2019-09-16T19:37:05.360Z,blockHash:
{{NoChM92yKRuJAB/jeLd1VnYn4DHiWIf071ACfic9uHc=}},entriesHash:
{{l05L0siKV14SDbuaYnH7uwXzUvqzIwUiRLXGbTyj/nY=}},previousBlockHash:
{{7kewBXhpdBc1cZKxhVmpoMHPUGOJtWQD0iY2LPfZkYA=}},entriesHashList:
[{{eRSwnmAM7WWANWdD5iG0yK+T4tDXyzUq6HZ/0fgLHos=}},{{mHVex/
```

```

yjHAWjFPpwhBuH2GKXmKJjK2FBa9faquUVNtg=}},
{{y5cCBz7p0AIUfsVQ1j0TqtE97b4b4oo1R0vnYyE5wWM=}},{{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/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": "[{{l3+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},
{{48CXG3ehPqsxCYd34EEa8Fso00RpWwA08010RJKf3Do=}},{{9UnwnKSQT0i3ge1JMVa
+tMIqCEDa0PTkWxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFctLufgPM6qXHyTNeCb1sCwcDaI=}},
{{Irb5fNhBrNEQ1VPhz1nGT/ZQPadSmgfdtMYcwkN0xoI=}},{{+3CwpYG/ytf/
vq9GidpzSx6JJiLXt1hMQWnNq0y3jfY=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT
+qE=}}]"
  }
}

```

Per ulteriori informazioni, consulta la sezione [Verifica dei dati in Amazon QLDB](#) nella Amazon QLDB Developer Guide.

- Per API i dettagli, consulta [GetBlock AWS CLI Command Reference](#).

## get-digest

Il seguente esempio di codice mostra come utilizzare `get-digest`.

### AWS CLI

Per ottenere un riassunto di un libro mastro

L'`get-digest` esempio seguente richiede un digest dal registro specificato all'ultimo blocco commesso nel diario.

```
aws qlldb get-digest \  
  --name vehicle-registration
```

Output:

```
{  
  "Digest": "6m6BMXobbJKpMhahwVthAEsN6awgnHK62Qq5McGP1Gk=",  
  "DigestTipAddress": {  
    "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:123}"  
  }  
}
```

Per ulteriori informazioni, consulta la sezione [Verifica dei dati in Amazon QLDB](#) nella Amazon QLDB Developer Guide.

- Per API i dettagli, consulta [GetDigest AWS CLI Command Reference](#).

## get-revision

Il seguente esempio di codice mostra come utilizzare `get-revision`.

### AWS CLI

Esempio 1: Per ottenere la revisione di un documento e una prova per la verifica utilizzando file di input

L'`get-revision` esempio seguente richiede un oggetto di dati di revisione e una bozza dal registro specificato. La richiesta riguarda un indirizzo digest tip, un ID del documento e un indirizzo di blocco della revisione specificati.

```
aws qlldb get-revision \  
  --name vehicle-registration \  
  --block-address file://myblockaddress.json \  
  --document-id JUJgkIcNbhS2goq8RqLuZ4 \  
  --digest-tip-address file://mydigesttipaddress.json
```

Contenuto di `myblockaddress.json`.

```
{  
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:100}"  
}
```

## Contenuto di `mydigesttipaddress.json`.

```
{
  "IonText": "{strandId:\\"KmA3ZZca7vAIiJAK9S5Iw1\\",sequenceNo:123}"
}
```

## Output:

```
{
  "Revision": {
    "IonText": "{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": "[[{{eRSwnmAM7WWANWd5iG0yK+T4tDXyzUq6HZ/0fgLHos=}},{{VV1rdaNuf+yJZVG1msM6gr2T52QvB08Lg+KgpjcnWAU=}},{{7kewBXhpdBc1cZKxhVmpoMhpUGOJtWQD0iY2LPfZkYA=}},{{13+EXs69K1+rehlqyWLkt+oHD1w4Zi9pCLW/t/mgTPM=}},{{48CXG3ehPqsxCYd34EEa8Fso00RpWWA08010RJKf3Do=}},{{9UnwnKSQT0i3ge1JMVa+tMIqCEDa0PTkWxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFcTlufgPM6qXHyTNECb1sCwcDaI=}},{{Irb5fNhBrNEQ1VPhz1nGT/ZQPadSmgfdtMYcwkN0xoI=}},{{+3CWpYG/ytf/vq9GidpzSx6JJiLXt1hMQWNnq0y3jfY=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT+qE=}}]"
  }
}
```

Per ulteriori informazioni, consulta la sezione [Verifica dei dati in Amazon QLDB](#) nella Amazon QLDB Developer Guide.

Esempio 2: per ottenere la revisione di un documento e una prova per la verifica utilizzando una sintassi abbreviata

L'`get-revision` seguente richiede un oggetto di dati di revisione e una bozza dal registro specificato utilizzando una sintassi abbreviata. La richiesta riguarda un indirizzo digest tip, un ID del documento e un indirizzo di blocco della revisione specificati.

```
aws qlldb get-revision \
```

```
--name vehicle-registration \
--block-address 'IonText="{strandId:\\"KmA3ZZca7vAIiJAK9S5IwL\\",sequenceNo:100}"'
\
--document-id JUJgkIcNbhS2goq8RqLuZ4 \
--digest-tip-address 'IonText="{strandId:\\"KmA3ZZca7vAIiJAK9S5IwL\\",sequenceNo:123}"'
```

Output:

```
{
  "Revision": {
    "IonText": "{blockAddress:{strandId:\\"KmA3ZZca7vAIiJAK9S5IwL\\",sequenceNo:100},hash:{{mHVex/yjHAWjFPpwhBuH2GKXmKjK2FBa9faquUVNtg=}},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=}},{{7kewBXhpdBc1cZKxhVmpoMHPUGOJtwQD0iY2LPfZkYA=}},{{13+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},{{48CXG3ehPqsxCYd34EEa8Fso00RpWwA08010RJKf3Do=}},{{9UnwnKSQT0i3ge1JMVa+tMIqCEDaOPTkwxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFcTlufgPM6qXHyTNECb1sCwcDaI=}},{{Irb5fNhBrNEQ1VPhzlnGT/ZQPadSmgfdtMYcwkN0xoI=}},{{+3CwpYG/ytf/vq9GidpzSx6JJiLXt1hMQWNnq0y3jfY=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUd02jB0AA0myZAT+qE=}}]"
  }
}
```

Per ulteriori informazioni, consulta la sezione [Verifica dei dati in Amazon QLDB](#) nella Amazon QLDB Developer Guide.

- Per API i dettagli, consulta [GetRevision AWS CLI](#) Command Reference.

## list-journal-kinesis-streams-for-ledger

Il seguente esempio di codice mostra come utilizzare `list-journal-kinesis-streams-for-ledger`.

## AWS CLI

Per elencare i flussi di diario per un libro mastro

L'`list-journal-kinesis-streams-for-ledger` seguente elenca i flussi di giornale per il libro mastro specificato.

```
aws qldb list-journal-kinesis-streams-for-ledger \  
  --ledger-name myExampleLedger
```

Output:

```
{  
  "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:qldb:us-east-1:123456789012:stream/  
myExampleLedger/7ISckqwe4y25YyHLzYUFaf",  
      "Status": "ACTIVE",  
      "KinesisConfiguration": {  
        "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-  
for-qldb",  
        "AggregationEnabled": true  
      },  
      "StreamName": "myExampleLedger-stream"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Streaming journal data from Amazon QLDB nell'Amazon QLDB Developer Guide](#).

- Per API i dettagli, consulta [ListJournalKinesisStreamsForLedger AWS CLI Command Reference](#).

## list-journal-s3-exports-for-ledger

Il seguente esempio di codice mostra come utilizzare `list-journal-s3-exports-for-ledger`.

### AWS CLI

Per elencare i lavori di esportazione delle riviste per un libro mastro

L'`list-journal-s3-exports-for-ledger` seguente elenca i lavori di esportazione delle scritture contabili per il libro contabile specificato.

```
aws qlldb list-journal-s3-exports-for-ledger \  
  --name myExampleLedger
```

Output:

```
{  
  "JournalS3Exports": [  
    {  
      "LedgerName": "myExampleLedger",  
      "ExclusiveEndTime": 1568847599.0,  
      "ExportCreationTime": 1568847801.418,  
      "S3ExportConfiguration": {  
        "Bucket": "awsExampleBucket",  
        "Prefix": "ledgerexport1/",  
        "EncryptionConfiguration": {  
          "ObjectEncryptionType": "SSE_S3"  
        }  
      },  
      "ExportId": "ADR2ONPKN5LINYGb4dp7yZ",  
      "RoleArn": "arn:aws:iam::123456789012:role/qlldb-s3-export",  
      "InclusiveStartTime": 1568764800.0,  
      "Status": "IN_PROGRESS"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Exporting Your Journal in Amazon QLDB nella Amazon QLDB Developer Guide](#).

- Per API i dettagli, consulta [ListJournalS3 ExportsForLedger](#) in AWS CLI Command Reference.

## list-journal-s3-exports

Il seguente esempio di codice mostra come usare `list-journal-s3-exports`.

### AWS CLI

Per elencare i lavori di esportazione delle riviste

L'`list-journal-s3-exports` seguente elenca i lavori di esportazione delle scritture contabili per tutti i libri contabili associati al AWS conto corrente e alla regione.

```
aws qldb list-journal-s3-exports
```

Output:

```
{
  "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
    }
]
}
```

Per ulteriori informazioni, consulta [Exporting Your Journal in Amazon QLDB nella Amazon QLDB Developer Guide](#).

- Per API i dettagli, consulta [ListJournalS3Exports](#) in AWS CLI Command Reference.

## list-ledgers

Il seguente esempio di codice mostra come utilizzare `list-ledgers`

### AWS CLI

Per elencare i registri disponibili

L'`list-ledgers` seguente elenca tutti i libri contabili associati al AWS conto corrente e alla regione.

```
aws qldb list-ledgers
```

Output:

```
{
  "Ledgers": [
    {
      "State": "ACTIVE",
      "CreationDateTime": 1568839243.951,
      "Name": "myExampleLedger"
    },
    {
      "State": "ACTIVE",
      "CreationDateTime": 1568839543.557,
      "Name": "myExampleLedger2"
    }
  ]
}
```

```
}
```

Per ulteriori informazioni, consulta [Basic Operations for Amazon QLDB Ledgers](#) nella Amazon QLDB Developer Guide.

- Per API i dettagli, consulta [ListLedgers AWS CLI Command Reference](#).

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag allegati a un libro mastro

L'`list-tags-for-resource` esempio seguente elenca tutti i tag allegati al registro specificato.

```
aws qldb list-tags-for-resource \  
  --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger
```

Output:

```
{  
  "Tags": {  
    "IsTest": "true",  
    "Domain": "Test"  
  }  
}
```

Per ulteriori informazioni, consulta [Tagging Amazon QLDB Resources](#) nell'Amazon QLDB Developer Guide.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## stream-journal-to-kinesis

Il seguente esempio di codice mostra come utilizzare `stream-journal-to-kinesis`.

### AWS CLI

Esempio 1: per trasmettere i dati del journal a Kinesis Data Streams utilizzando file di input

L'operazione `stream-journal-to-kinesis` seguente crea un flusso di dati del diario entro un intervallo di data e ora specificato da un registro con lo stesso nome. `myExampleLedger` Il flusso invia i dati a un flusso di dati Amazon Kinesis specificato.

```
aws qldb 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
```

Contenuto di `my-kinesis-config.json`.

```
{  
  "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-qldb",  
  "AggregationEnabled": true  
}
```

Output:

```
{  
  "StreamId": "7ISckqwe4y25YyHLzYUFaf"  
}
```

Per ulteriori informazioni, consulta [Streaming journal data from Amazon QLDB nell'Amazon QLDB Developer Guide](#).

Esempio 2: Per trasmettere i dati del journal a Kinesis Data Streams utilizzando una sintassi abbreviata

L'operazione `stream-journal-to-kinesis` seguente crea un flusso di dati del diario entro un intervallo di data e ora specificato da un registro con lo stesso nome. `myExampleLedger` Il flusso invia i dati a un flusso di dati Amazon Kinesis specificato.

```
aws qldb 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 file://my-kinesis-config.json
```

```
--kinesis-configuration StreamArn=arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-qldb,AggregationEnabled=true
```

Output:

```
{  
  "StreamId": "7ISckqwe4y25YyHLzYUFAf"  
}
```

Per ulteriori informazioni, consulta [Streaming journal data from Amazon QLDB nell'Amazon QLDB Developer Guide](#).

- Per API i dettagli, consulta [StreamJournalToKinesis AWS CLI Command Reference](#).

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare un libro mastro

L'`tag-resource` esempio seguente aggiunge un set di tag a un registro specificato.

```
aws qlldb tag-resource \  
  --resource-arn arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger \  
  --tags IsTest=true,Domain=Test
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Amazon QLDB Resources](#) nell'Amazon QLDB Developer Guide.

- Per API i dettagli, consulta [TagResource AWS CLI Command Reference](#).

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una risorsa

L'`untag-resource` seguente rimuove i tag con le chiavi dei tag specificati da un registro specificato.

```
aws qlldb untag-resource \  
  --resource-arn arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger \  
  --tag-keys IsTest Domain
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Amazon QLDB Resources](#) nell'Amazon QLDB Developer Guide.

- Per API i dettagli, consulta [UntagResource AWS CLI](#) Command Reference.

## update-ledger-permissions-mode

Il seguente esempio di codice mostra come utilizzare `update-ledger-permissions-mode`.

### AWS CLI

Esempio 1: Aggiornare la modalità di autorizzazione di un registro su STANDARD

L'`update-ledger-permissions-mode` seguente assegna la modalità di STANDARD autorizzazione al registro specificato.

```
aws qlldb update-ledger-permissions-mode \  
  --name myExampleLedger \  
  --permissions-mode STANDARD
```

Output:

```
{  
  "Name": "myExampleLedger",  
  "Arn": "arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger",  
  "PermissionsMode": "STANDARD"  
}
```

Esempio 2: Aggiornare la modalità di autorizzazione di un libro mastro a `_ALLOW_ALL`

L'`update-ledger-permissions-mode` seguente assegna la modalità di `ALLOW_ALL` autorizzazione al registro specificato.

```
aws qlldb update-ledger-permissions-mode \  
  --name myExampleLedger \  
  --permissions-mode ALLOW_ALL
```

Output:

```
{  
  "Name": "myExampleLedger",  
  "Arn": "arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger",  
  "PermissionsMode": "ALLOW_ALL"  
}
```

Per ulteriori informazioni, consulta [Basic Operations for Amazon QLDB Ledgers](#) nella Amazon QLDB Developer Guide.

- Per API i dettagli, consulta [UpdateLedgerPermissionsMode AWS CLI Command Reference](#).

## update-ledger

Il seguente esempio di codice mostra come utilizzare `update-ledger`.

### AWS CLI

Esempio 1: per aggiornare la proprietà di protezione dall'eliminazione di un libro mastro

L'`update-ledger` seguente aggiorna il registro specificato per disabilitare la funzionalità di protezione dall'eliminazione.

```
aws qlldb update-ledger \  
  --name myExampleLedger \  
  --no-deletion-protection
```

Output:

```
{  
  "CreationDateTime": 1568839243.951,  
  "Arn": "arn:aws:qlldb:us-west-2:123456789012:ledger/myExampleLedger",  
  "DeletionProtection": false,  
  "Name": "myExampleLedger",  
  "State": "ACTIVE"
```

```
}
```

Esempio 2: aggiornare la AWS KMS chiave di un libro mastro con una chiave gestita dal cliente

L'update-ledgeresempio seguente aggiorna il registro specificato per utilizzare una KMS chiave gestita dal cliente per la crittografia inattiva.

```
aws qlldb update-ledger \  
  --name myExampleLedger \  
  --kms-key arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "CreationDateTime": 1568839243.951,  
  "Arn": "arn:aws:qlldb: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"  
  }  
}
```

Esempio 3: aggiornare la AWS KMS chiave di un libro mastro con una chiave proprietaria AWS

L'update-ledgeresempio seguente aggiorna il registro specificato per utilizzare una KMS chiave AWS proprietaria per la crittografia a riposo.

```
aws qlldb update-ledger \  
  --name myExampleLedger \  
  --kms-key AWS_OWNED_KMS_KEY
```

Output:

```
{  
  "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"  
}  
}
```

Per ulteriori informazioni, consulta [Basic Operations for Amazon QLDB Ledgers](#) nella Amazon QLDB Developer Guide.

- Per API i dettagli, consulta [UpdateLedger AWS CLI](#) Command Reference.

## RDSEsempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con AmazonRDS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **add-option-to-option-group**

Il seguente esempio di codice mostra come utilizzare `add-option-to-option-group`.

#### AWS CLI

Per aggiungere un'opzione a un gruppo di opzioni

L'`add-option-to-option-group` esempio seguente aggiunge un'opzione al gruppo di opzioni specificato.



```
aws rds add-option-to-option-group \
  --option-group-name myoptiongroup \
  --options OptionName=OEM,Port=5500,DBSecurityGroupMemberships=default \
  --apply-immediately
```

## Output:

```
{
  "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"
}
}

```

Per ulteriori informazioni, consulta [Aggiungere un'opzione a un gruppo di opzioni](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [AddOptionToOptionGroup AWS CLI](#) Command Reference.

## add-role-to-db-cluster

Il seguente esempio di codice mostra come utilizzare `add-role-to-db-cluster`.

### AWS CLI

Per associare un ruolo AWS Identity and Access Management (IAM) a un cluster DB

L'`add-role-to-db-cluster` esempio seguente associa un ruolo a un cluster DB.

```
aws rds add-role-to-db-cluster \  
  --db-cluster-identifier mydbcluster \  
  --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Associare un IAM ruolo a un cluster Amazon Aurora SQL My DB](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta Command [AddRoleToDbCluster](#) Reference AWS CLI .

## add-role-to-db-instance

Il seguente esempio di codice mostra come utilizzare `add-role-to-db-instance`.

### AWS CLI

Per associare un ruolo AWS Identity and Access Management (IAM) a un'istanza DB

L'`add-role-to-db-instance` esempio seguente aggiunge il ruolo a un'istanza Oracle DB denominata `test-instance`.

```
aws rds add-role-to-db-instance \  
  --db-instance-identifier test-instance \  
  --feature-name S3_INTEGRATION \  
  --role-arn arn:aws:iam::111122223333:role/rds-s3-integration-role
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Prerequisiti per l'integrazione di Amazon RDS Oracle con Amazon S3](#) nella RDS Amazon User Guide.

- Per API i dettagli, consulta AWS CLI Command [AddRoleToDbInstanceReference](#).

## add-source-identifier-to-subscription

Il seguente esempio di codice mostra come utilizzare `add-source-identifier-to-subscription`.

### AWS CLI

Per aggiungere un identificatore di origine a un abbonamento

L'`add-source-identifier` esempio seguente aggiunge un altro identificatore di origine a un abbonamento esistente.

```
aws rds add-source-identifier-to-subscription \  
  --subscription-name my-instance-events \  
  --source-identifier test-instance-repl
```

Output:

```
{  
  "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"  
    ]  
  }  
}
```

- Per API i dettagli, vedere [AddSourceIdentifierToSubscription](#) in AWS CLI Command Reference.

## add-tags-to-resource

Il seguente esempio di codice mostra come utilizzare `add-tags-to-resource`.

### AWS CLI

Per aggiungere tag a una risorsa

L'`add-tags-to-resource` esempio seguente aggiunge tag a un RDS database.

```
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\"}]"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Amazon RDS Resources](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [AddTagsToResource AWS CLI](#) Command Reference.

## apply-pending-maintenance-action

Il seguente esempio di codice mostra come utilizzare `apply-pending-maintenance-action`.

### AWS CLI

Esecuzione di azioni di manutenzione in attesa

L'`apply-pending-maintenance-action` esempio seguente applica le azioni di manutenzione in sospeso per un cluster DB.

```
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
```

Output:

```
{
  "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"
      }
    ]
  }
}
```

Per ulteriori informazioni, consulta [Manutenzione di un'istanza DB](#) nella Amazon RDS User Guide e [Manutenzione di un cluster DB Amazon Aurora](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta AWS CLI Command [ApplyPendingMaintenanceAction](#) Reference.

## authorize-db-security-group-ingress

Il seguente esempio di codice mostra come utilizzare `authorize-db-security-group-ingress`.

### AWS CLI

Per associare un ruolo AWS Identity and Access Management (IAM) a un'istanza DB

L'`authorize-db-security-group-ingress` seguente configura il gruppo di sicurezza predefinito con una regola di ingresso per l'intervallo CIDR IP 192.0.2.0/24.

```
aws rds authorize-db-security-group-ingress \
  --db-security-group-name default \
  --cidrip 192.0.2.0/24
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta la sezione [Autorizzazione dell'accesso alla rete a un gruppo di sicurezza DB da un intervallo IP](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [AuthorizeDbSecurityGroupIngress AWS CLI Command Reference](#).

## backtrack-db-cluster

Il seguente esempio di codice mostra come utilizzare `backtrack-db-cluster`.

### AWS CLI

Per eseguire il backtrack di un cluster Aurora DB

L'`backtrack-db-cluster` esempio seguente esegue il backtrack del cluster DB `sample-cluster` specificato al 19 marzo 2018 alle 10.00.

```
aws rds backtrack-db-cluster --db-cluster-identifier sample-cluster --backtrack-to 2018-03-19T10:00:00+00:00
```

Questo comando genera un JSON blocco che riconosce la modifica alla risorsa. RDS

- Per API i dettagli, vedere [BacktrackDbCluster](#) in AWS CLI Command Reference.

## cancel-export-task

Il seguente esempio di codice mostra come utilizzare `cancel-export-task`.

### AWS CLI

Per annullare l'esportazione di uno snapshot in Amazon S3

L'command `aws rds cancel-export-task` seguente annulla un'attività di esportazione in corso che sta esportando uno snapshot in Amazon S3.

```
aws rds cancel-export-task \  
  --export-task-identifier my-s3-export-1
```

Output:

```
{  
  "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  
}
```

Per ulteriori informazioni, consulta [Annullamento di un'attività di esportazione di snapshot](#) nella Amazon RDS User Guide o Annullamento di un'attività di esportazione di snapshot nella Guida per l'utente [di Amazon Aurora](#).

- Per API i dettagli, consulta Command Reference. [CancelExportTask](#) AWS CLI

## copy-db-cluster-parameter-group

Il seguente esempio di codice mostra come utilizzare `copy-db-cluster-parameter-group`.

AWS CLI

Per copiare un gruppo di parametri del cluster DB

L'command `aws rds copy-db-cluster-parameter-group` seguente crea una copia di un gruppo di parametri del cluster DB.

```
aws rds copy-db-cluster-parameter-group \  
  --source-db-cluster-parameter-group-identifier mydbclusterpg \  
  --target-db-cluster-parameter-group-identifier mydbclusterpg-copy
```



```
--target-db-cluster-parameter-group-identifier mydbclusterpgcopy \
--target-db-cluster-parameter-group-description "Copy of mydbclusterpg parameter
group"
```

Output:

```
{
  "DBClusterParameterGroup": {
    "DBClusterParameterGroupName": "mydbclusterpgcopy",
    "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterpgcopy",
    "DBParameterGroupFamily": "aurora-mysql5.7",
    "Description": "Copy of mydbclusterpg parameter group"
  }
}
```

Per ulteriori informazioni, [consulta Copiare un gruppo di parametri del cluster DB nella Guida](#) per gli utenti di Amazon Aurora.

- Per API i dettagli, consulta AWS CLI Command [CopyDbClusterParameterGroupReference](#).

## copy-db-cluster-snapshot

Il seguente esempio di codice mostra come utilizzare `copy-db-cluster-snapshot`.

AWS CLI

Per copiare un'istantanea del cluster DB

L'`copy-db-cluster-snapshot`esempio seguente crea una copia di uno snapshot del cluster DB, inclusi i relativi tag.

```
aws rds copy-db-cluster-snapshot \
  --source-db-cluster-snapshot-identifier arn:aws:rds:us-
east-1:123456789012:cluster-snapshot:rds:myaurora-2019-06-04-09-16
  --target-db-cluster-snapshot-identifier myclustersnapshotcopy \
  --copy-tags
```

Output:

```
{
  "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
  }
}

```

Per ulteriori informazioni, consulta [Copiare uno snapshot nella Guida](#) per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [CopyDbClusterSnapshot](#) Reference AWS CLI .

## copy-db-parameter-group

Il seguente esempio di codice mostra come utilizzare `copy-db-parameter-group`.

### AWS CLI

Per copiare un gruppo di parametri del cluster DB

L'`copy-db-parameter-group`esempio seguente crea una copia di un gruppo di parametri DB.

```
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"
```

Output:

```
{  
  "DBParameterGroup": {  
    "DBParameterGroupName": "mydbpgcopy",  
    "DBParameterGroupArn": "arn:aws:rds:us-east-1:814387698303:pg:mydbpgcopy",  
    "DBParameterGroupFamily": "mysql5.7",  
    "Description": "Copy of mydbpg parameter group"  
  }  
}
```

Per ulteriori informazioni, [consulta Copiare un gruppo di parametri DB](#) nella Amazon RDS Users Guide.

- Per API i dettagli, consulta [CopyDbParameterGroup AWS CLI Command Reference](#).

## copy-db-snapshot

Il seguente esempio di codice mostra come utilizzare `copy-db-snapshot`.

### AWS CLI

Per copiare un'istantanea del DB

L'`copy-db-snapshot` seguente crea una copia di uno snapshot DB.

```
aws rds copy-db-snapshot \  
--source-db-snapshot-identifier rds:database-mysql-2019-06-06-08-38  
--target-db-snapshot-identifier mydbsnapshotcopy
```

Output:

```
{  
  "DBSnapshot": {  
    "VpcId": "vpc-6594f31c",  
    "Status": "creating",  
    "Encrypted": true,  
  }  
}
```

```

    "SourceDBSnapshotIdentifier": "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",
    "DBInstanceIdentifier": "database-mysql",
    "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
    "AvailabilityZone": "us-east-1f",
    "PercentProgress": 0,
    "AllocatedStorage": 100,
    "DBSnapshotIdentifier": "mydbsnapshotcopy"
  }
}

```

Per ulteriori informazioni, consulta [Copiare uno snapshot](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [CopyDbSnapshotReference](#).

## copy-option-group

Il seguente esempio di codice mostra come utilizzare `copy-option-group`.

### AWS CLI

Per copiare un gruppo di opzioni

L'`copy-option-group` esempio seguente crea una copia di un gruppo di opzioni.

```

aws rds copy-option-group \
  --source-option-group-identifier myoptiongroup \
  --target-option-group-identifier new-option-group \

```

```
--target-option-group-description "My option group copy"
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Making a Copy of an Option Group](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [CopyOptionGroup AWS CLI Command Reference](#).

## create-blue-green-deployment

Il seguente esempio di codice mostra come utilizzare `create-blue-green-deployment`.

AWS CLI

Esempio 1: creare una distribuzione blu/verde per un'istanza RDS for My DB SQL

L'`create-blue-green-deployment` esempio seguente crea una distribuzione blu/verde per un'istanza My DB. SQL

```
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
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Creare una distribuzione blu/verde](#) nella Amazon RDS User Guide.

Esempio 2: creare una distribuzione blu/verde per un cluster Aurora My DB SQL

L'create-blue-green-deploymentesempio seguente crea una distribuzione blu/verde per un cluster Aurora My DB. SQL

```
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
```

Output:

```
{  
  "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"
}
}

```

Per ulteriori informazioni, consulta [Creare una distribuzione blu/verde](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [CreateBlueGreenDeploymentReference](#) AWS CLI .

## **create-db-cluster-endpoint**

Il seguente esempio di codice mostra come utilizzare `create-db-cluster-endpoint`.



## AWS CLI

Per creare un endpoint di cluster DB personalizzato

L'`create-db-cluster-endpoint` seguente crea un endpoint cluster DB personalizzato e lo associa al cluster Aurora DB specificato.

```
aws rds create-db-cluster-endpoint \  
  --db-cluster-endpoint-identifier mycustomendpoint \  
  --endpoint-type reader \  
  --db-cluster-identifier mydbcluster \  
  --static-members dbinstance1 dbinstance2
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Amazon Aurora Connection Management](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateDbClusterEndpoint](#) Reference.

## `create-db-cluster-parameter-group`

Il seguente esempio di codice mostra come utilizzare `create-db-cluster-parameter-group`.

## AWS CLI

Per creare un gruppo di parametri del cluster DB

L'`create-db-cluster-parameter-group` seguente crea un gruppo di parametri del cluster DB.

```
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"
```

Output:

```
{  
  "DBClusterParameterGroup": {  
    "DBClusterParameterGroupName": "mydbclusterparametergroup",  
    "DBParameterGroupFamily": "aurora5.6",  
    "Description": "My new cluster parameter group",  
    "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-  
pg:mydbclusterparametergroup"  
  }  
}
```

Per ulteriori informazioni, consulta [Creazione di un gruppo di parametri del cluster DB](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta [CreateDbClusterParameterGroup AWS CLI Command Reference](#).

## `create-db-cluster-snapshot`

Il seguente esempio di codice mostra come utilizzare `create-db-cluster-snapshot`.

### AWS CLI

Per creare uno snapshot del cluster DB

L'`create-db-cluster-snapshot` seguente crea uno snapshot del cluster DB.

```
aws rds create-db-cluster-snapshot \  
  --db-cluster-identifier mydbcluster \  
  --db-cluster-snapshot-identifier mydbclustersnapshot
```

**Output:**

```
{
  "DBClusterSnapshot": {
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1b",
      "us-east-1e"
    ],
    "DBClusterSnapshotIdentifier": "mydbclustersnapshot",
    "DBClusterIdentifier": "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
  }
}
```

Per ulteriori informazioni, consulta [Creazione di uno snapshot del cluster DB](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta AWS CLI Command [CreateDbClusterSnapshotReference](#).

**create-db-cluster**

Il seguente esempio di codice mostra come utilizzare `create-db-cluster`.

**AWS CLI**

Esempio 1: creare un cluster DB compatibile con My SQL 5.7

L'create-db-cluster esempio seguente crea un cluster DB SQL compatibile con My 5.7 utilizzando la versione del motore predefinita. Sostituisci la password di esempio secret99 con una password sicura. Quando usi la console per creare un cluster DB, Amazon crea RDS automaticamente l'istanza Writer DB per il tuo cluster DB. Tuttavia, quando utilizzi il AWS CLI per creare un cluster DB, devi creare esplicitamente l'istanza Writer DB per il tuo cluster DB utilizzando il create-db-instance AWS CLI comando.

```
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
```

Output:

```
{  
  "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
  }
}

```

## Esempio 2: per creare un cluster DB compatibile con Postgre --compatible SQL

L'`create-db-cluster` seguente crea un cluster DB SQL compatibile con Postgre utilizzando la versione del motore predefinita. Sostituisci la password di esempio `secret99` con una password sicura. Quando usi la console per creare un cluster DB, Amazon crea RDS automaticamente l'istanza Writer DB per il tuo cluster DB. Tuttavia, quando utilizzi il AWS CLI per creare un cluster DB, devi creare esplicitamente l'istanza Writer DB per il tuo cluster DB utilizzando il `create-db-instance` AWS CLI comando.

```

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

```

Output:

```

{
  "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"
  }
```

```
}
```

Per ulteriori informazioni, consulta [Creazione di un cluster Amazon Aurora DB](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta AWS CLI Command [CreateDbCluster](#)Reference.

## **create-db-instance-read-replica**

Il seguente esempio di codice mostra come utilizzare `create-db-instance-read-replica`.

### AWS CLI

Per creare un'istanza DB, leggi la replica

Questo esempio crea una replica di lettura di un'istanza DB esistente denominata. `test-instance` La replica di lettura è denominata. `test-instance-repl`

```
aws rds create-db-instance-read-replica \  
  --db-instance-identifier test-instance-repl \  
  --source-db-instance-identifier test-instance
```

Output:

```
{  
  "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...  
  }  
}
```

- Per API i dettagli, vedere [CreateDbInstanceReadReplica](#)in AWS CLI Command Reference.

## **create-db-instance**

Il seguente esempio di codice mostra come utilizzare `create-db-instance`.

## AWS CLI

Per creare un'istanza DB

L'create-db-instanceesempio seguente utilizza le opzioni richieste per avviare una nuova istanza DB.

```
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
```

Output:

```
{  
  "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-5555EXAMPLE44444444EXAMPLE",
  "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": []
}
}

```

Per ulteriori informazioni, consulta [Creating an Amazon RDS DB Instance](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [C createDBInstance](#) in AWS CLI Command Reference.

## create-db-parameter-group

Il seguente esempio di codice mostra come usare `create-db-parameter-group`.

### AWS CLI

Per creare un gruppo di parametri DB

L'`create-db-parameter-group`esempio seguente crea un gruppo di parametri DB.

```

aws rds create-db-parameter-group \
  --db-parameter-group-name mydbparametergroup \
  --db-parameter-group-family MySQL5.6 \
  --description "My new parameter group"

```

Output:

```
{
  "DBParameterGroup": {
    "DBParameterGroupName": "mydbparametergroup",
    "DBParameterGroupFamily": "mysql5.6",
    "Description": "My new parameter group",
    "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:mydbparametergroup"
  }
}
```

Per ulteriori informazioni, consulta [Creating a DB Parameter Group](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [C reateDBParameter Group](#) in AWS CLI Command Reference.

## create-db-proxy-endpoint

Il seguente esempio di codice mostra come utilizzare `create-db-proxy-endpoint`.

### AWS CLI

Per creare un endpoint proxy DB per un RDS database

L'`create-db-proxy-endpoint` seguente crea un endpoint proxy DB.

```
aws rds create-db-proxy-endpoint \
  --db-proxy-name proxyExample \
  --db-proxy-endpoint-name "proxyep1" \
  --vpc-subnet-ids subnetgroup1 subnetgroup2
```

Output:

```
{
  "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
  }
}

```

Per ulteriori informazioni, consulta [Creazione di un endpoint proxy](#) nella Amazon RDS User Guide e [Creazione di un endpoint proxy](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [CreateDbProxyEndpoint](#) Reference AWS CLI .

## create-db-proxy

Il seguente esempio di codice mostra come utilizzare `create-db-proxy`.

### AWS CLI

Per creare un proxy DB per un RDS database

L'esempio seguente crea un proxy DB.

```

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="MYSQL"
  \
  --role-arn arn:aws:iam::123456789123:role/ProxyRole \
  --vpc-subnet-ids subnetgroup1 subnetgroup2

```

Output:

```
{
```

```

"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:proxyssecret1-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"
}
}

```

Per ulteriori informazioni, consulta [Creazione di un RDS proxy](#) nella Amazon RDS User Guide e [Creazione di un RDS proxy](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta [CreateDbProxy AWS CLI Command Reference](#).

## create-db-security-group

Il seguente esempio di codice mostra come utilizzare `create-db-security-group`.

## AWS CLI

Per creare un gruppo di sicurezza Amazon RDS DB

Il `create-db-security-group` comando seguente crea un nuovo gruppo di sicurezza Amazon RDS DB:

```
aws rds create-db-security-group --db-security-group-name mysecgroup --db-security-group-description "My Test Security Group"
```

Nell'esempio, il nuovo gruppo di sicurezza DB ha un nome `mysecgroup` e una descrizione.

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [CreateDbSecurityGroup](#) in AWS CLI Command Reference.

## `create-db-shard-group`

Il seguente esempio di codice mostra come utilizzare `create-db-shard-group`.

## AWS CLI

Esempio 1: creare un cluster DB primario Aurora SQL Postgre

L'`create-db-cluster` esempio seguente crea un cluster DB SQL SQL primario Aurora Postgre compatibile con Aurora Serverless v2 e Aurora Limitless Database.

```
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
```

Output:

```
{  
  "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",
    "IOOptimizedNextAllowedModificationTime":
"2024-03-21T16:24:07.781000+00:00",
    "LimitlessDatabase": {
      "Status": "not-in-use",
      "MinRequiredACU": 96.0
    }
  }
}

```

## Esempio 2: creare l'istanza DB primaria (writer)

L'create-db-instanceesempio seguente crea un'istanza DB primaria (writer) Aurora Serverless v2. Quando usi la console per creare un cluster DB, Amazon crea RDS automaticamente l'istanza Writer DB per il tuo cluster DB. Tuttavia, quando utilizzi il AWS CLI per creare un cluster DB, devi creare esplicitamente l'istanza Writer DB per il tuo cluster DB utilizzando il create-db-instance AWS CLI comando.

```

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
```

Output:

```
{  
  "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
  }
}

```

### Esempio 3: per creare il gruppo di shard DB

L'create-db-shard-groupesempio seguente crea un gruppo di shard DB nel cluster DB primario Aurora SQL Postgre.

```

aws rds create-db-shard-group \
  --db-shard-group-identifier my-db-shard-group \
  --db-cluster-identifier my-sv2-cluster \
  --max-acu 768

```

### Output:

```

{
  "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
  "DBShardGroupIdentifier": "my-db-shard-group",
  "DBClusterIdentifier": "my-sv2-cluster",
  "MaxACU": 768.0,
  "ComputeRedundancy": 0,
  "Status": "creating",
  "PubliclyAccessible": false,
  "Endpoint": "my-sv2-cluster.limitless-cekyceexample.us-east-2.rds.amazonaws.com"
}

```

```
}
```

Per ulteriori informazioni, consulta [Using Aurora Serverless v2](#) nella Amazon Aurora User Guide.

- Per i API dettagli, consulta [CreateDbShardGroup](#) Command Reference.AWS CLI

## create-db-snapshot

Il seguente esempio di codice mostra come utilizzare `create-db-snapshot`.

### AWS CLI

Per creare un'istantanea del DB

L'`create-db-snapshot` esempio seguente crea un'istantanea del DB.

```
aws rds create-db-snapshot \  
  --db-instance-identifier database-mysql \  
  --db-snapshot-identifier mydbsnapshot
```

Output:

```
{  
  "DBSnapshot": {  
    "DBSnapshotIdentifier": "mydbsnapshot",  
    "DBInstanceIdentifier": "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"
  }
}

```

Per ulteriori informazioni, consulta [Creating a DB Snapshot](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [C createDBSnapshot](#) in AWS CLI Command Reference.

## create-db-subnet-group

Il seguente esempio di codice mostra come usare `create-db-subnet-group`.

### AWS CLI

Per creare un gruppo di sottoreti DB

L'esempio seguente crea un gruppo di sottoreti DB chiamato `mysubnetgroup` utilizzando sottoreti esistenti.

```

aws rds create-db-subnet-group \
  --db-subnet-group-name mysubnetgroup \
  --db-subnet-group-description "test DB subnet group" \
  --subnet-ids
'["subnet-0a1dc4e1a6f123456","subnet-070dd7ecb3aaaaaaaa","subnet-00f5b198bc0abcdef"]'

```

Output:

```

{
  "DBSubnetGroup": {
    "DBSubnetGroupName": "mysubnetgroup",
    "DBSubnetGroupDescription": "test DB subnet group",
    "VpcId": "vpc-0f08e7610a1b2c3d4",
    "SubnetGroupStatus": "Complete",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-070dd7ecb3aaaaaaaa",
        "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"
}
}

```

Per ulteriori informazioni, consulta [Creating a DB Instance in a VPC](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [CreateDbSubnetGroup AWS CLI Command Reference](#).

## create-event-subscription

Il seguente esempio di codice mostra come utilizzare `create-event-subscription`.

### AWS CLI

Per creare una sottoscrizione a un evento

L'esempio seguente crea una sottoscrizione per gli eventi di backup e ripristino per le istanze DB nell'AWS account corrente. Le notifiche vengono inviate a un argomento di Amazon Simple Notification Service, specificato da `--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
```

Output:

```
{
  "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
  }
}
```

- Per API i dettagli, consulta [CreateEventSubscription AWS CLI Command Reference](#).

## create-global-cluster

Il seguente esempio di codice mostra come utilizzare `create-global-cluster`.

AWS CLI

Per creare un cluster DB globale

L'esempio seguente crea un nuovo cluster DB globale SQL compatibile con Aurora My.

```
aws rds create-global-cluster \
  --global-cluster-identifier myglobalcluster \
  --engine aurora-mysql
```

Output:

```
{
```

```

    "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": []
    }
  }
}

```

Per ulteriori informazioni, consulta [Creazione di un database globale Aurora](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta AWS CLI Command [CreateGlobalCluster](#)Reference.

## create-option-group

Il seguente esempio di codice mostra come utilizzare `create-option-group`.

### AWS CLI

Per creare un gruppo di RDS opzioni Amazon

Il `create-option-group` comando seguente crea un nuovo gruppo di RDS opzioni Amazon per la Oracle Enterprise Edition versione 11.2`, is named ``MyOptionGroup e include una descrizione.

```

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"

```

Output:

```

{
  "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"
  }
}

```

- Per API i dettagli, consulta [CreateOptionGroup AWS CLI Command Reference](#).

## delete-blue-green-deployment

Il seguente esempio di codice mostra come utilizzare `delete-blue-green-deployment`.

### AWS CLI

Esempio 1: eliminare risorse in un ambiente verde per un'istanza RDS for My SQL DB

L'`delete-blue-green-deployment` esempio seguente elimina le risorse in un ambiente verde per un'istanza RDS for My SQL DB.

```

aws rds delete-blue-green-deployment \
  --blue-green-deployment-identifier bgd-v53303651eexfake \
  --delete-target

```

Output:

```

{
  "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"
}
```

```
}
```

Per ulteriori informazioni, consulta [Eliminazione di una distribuzione blu/verde](#) nella Amazon RDS User Guide.

Esempio 2: eliminare risorse in ambiente verde per un cluster Aurora My DB SQL

L'`delete-blue-green-deployment` seguente elimina le risorse in un ambiente verde per un cluster Aurora SQL My DB.

```
aws rds delete-blue-green-deployment \  
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake \  
  --delete-target
```

Output:

```
{  
  "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-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": "DELETING",
    "CreateTime": "2022-02-25T21:12:00.288000+00:00",
    "DeleteTime": "2022-02-25T22:29:11.336000+00:00"
  }
}

```

Per ulteriori informazioni, consulta [Eliminazione di una distribuzione blu/verde](#) nella Guida per l'utente di Amazon Aurora.

- Per i API dettagli, consulta [DeleteBlueGreenDeployment](#) Command Reference.AWS CLI

## delete-db-cluster-endpoint

Il seguente esempio di codice mostra come utilizzare `delete-db-cluster-endpoint`.

### AWS CLI

Per eliminare un endpoint del cluster DB personalizzato

L'`delete-db-cluster-endpoint`esempio seguente elimina l'endpoint del cluster DB personalizzato specificato.

```

aws rds delete-db-cluster-endpoint \
  --db-cluster-endpoint-identifier mycustomendpoint

```

Output:

```

{
  "DBClusterEndpointIdentifier": "mycustomendpoint",
  "DBClusterIdentifier": "mydbcluster",
  "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",
  "Endpoint": "mycustomendpoint.cluster-custom-cnpexample.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"
}
```

Per ulteriori informazioni, consulta [Amazon Aurora Connection Management](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteDbClusterEndpointReference](#).

## **delete-db-cluster-parameter-group**

Il seguente esempio di codice mostra come utilizzare `delete-db-cluster-parameter-group`.

### AWS CLI

Per eliminare un gruppo di parametri del cluster DB

L'`delete-db-cluster-parameter-group`esempio seguente elimina il gruppo di parametri del cluster DB specificato.

```
aws rds delete-db-cluster-parameter-group \
  --db-cluster-parameter-group-name mydbclusterparametergroup
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Working with DB Parameter Groups e DB Cluster Parameter Groups](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta [DeleteDbClusterParameterGroup AWS CLI Command Reference](#).

## **delete-db-cluster-snapshot**

Il seguente esempio di codice mostra come utilizzare `delete-db-cluster-snapshot`.

### AWS CLI

Per eliminare uno snapshot del cluster DB

L'`delete-db-cluster-snapshot`esempio seguente elimina lo snapshot del cluster DB specificato.

```
aws rds delete-db-cluster-snapshot \  
--db-cluster-snapshot-identifier mydbclustersnapshot
```

Output:

```
{  
  "DBClusterSnapshot": {  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1b",  
      "us-east-1e"  
    ],  
    "DBClusterSnapshotIdentifier": "mydbclustersnapshot",  
    "DBClusterIdentifier": "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  
  }  
}
```

Per ulteriori informazioni, consulta [Eliminazione di uno snapshot nella Guida](#) per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [DeleteDbClusterSnapshotReference](#) AWS CLI .

## delete-db-cluster

Il seguente esempio di codice mostra come utilizzare `delete-db-cluster`.

## AWS CLI

Esempio 1: per eliminare un'istanza DB in un cluster DB

L'`delete-db-instance` seguente elimina l'istanza DB finale in un cluster DB. Non è possibile eliminare un cluster DB se contiene istanze DB che non sono in stato di eliminazione. Non è possibile scattare un'istantanea finale quando si elimina un'istanza DB in un cluster di database.

```
aws rds delete-db-instance \  
  --db-instance-identifier database-3
```

Output:

```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "database-3",  
    "DBInstanceClass": "db.r4.large",  
    "Engine": "aurora-postgresql",  
    "DBInstanceStatus": "deleting",  
  
    ...output omitted...  
  }  
}
```

Per ulteriori informazioni, consulta [Eliminazione di un'istanza DB in un cluster DB Aurora](#) nella Guida per l'utente di Amazon Aurora.

Esempio 2: Per eliminare un cluster DB

L'`delete-db-cluster` seguente elimina il cluster DB denominato `mycluster` e crea un'istantanea finale denominata `mycluster-final-snapshot`. Lo stato del cluster DB è disponibile durante l'acquisizione dell'istantanea. Per seguire l'avanzamento della cancellazione, usa il `describe-db-clusters` CLI comando.

```
aws rds delete-db-cluster \  
  --db-cluster-identifier mycluster \  
  --no-skip-final-snapshot \  
  --final-db-snapshot-identifier mycluster-final-snapshot
```



## Output:

```
{
  "DBCluster": {
    "AllocatedStorage": 20,
    "AvailabilityZones": [
      "eu-central-1b",
      "eu-central-1c",
      "eu-central-1a"
    ],
    "BackupRetentionPeriod": 7,
    "DBClusterIdentifier": "mycluster",
    "DBClusterParameterGroup": "default.aurora-postgresql10",
    "DBSubnetGroup": "default-vpc-aa11bb22",
    "Status": "available",

    ...output omitted...

  }
}
```

Per ulteriori informazioni, consulta [Aurora Clusters with a Single DB Instance](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta Command [DeleteDbCluster](#)Reference AWS CLI .

**delete-db-instance-automated-backup**

Il seguente esempio di codice mostra come utilizzare `delete-db-instance-automated-backup`.

## AWS CLI

Per eliminare un backup automatico replicato da una regione

L'`delete-db-instance-automated-backup`esempio seguente elimina il backup automatico con l'Amazon Resource Name specificato (ARN).

```
aws rds delete-db-instance-automated-backup \
  --db-instance-automated-backups-arn "arn:aws:rds:us-west-2:123456789012:auto-backup:ab-jkib2gfg5rv7replzadausbrktni2bn4example"
```

## Output:

```
{
  "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-jkib2gfg5rv7replzadtausbrktni2bn4example"
  }
}
```

Per ulteriori informazioni, consulta [Eliminazione dei backup replicati](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta Command [DeleteDbInstanceAutomatedBackup](#) Reference AWS CLI .

## delete-db-instance

Il seguente esempio di codice mostra come utilizzare `delete-db-instance`.

### AWS CLI

Per eliminare un'istanza DB

L'`delete-db-instance` esempio seguente elimina l'istanza DB specificata dopo aver creato uno snapshot DB finale denominato `test-instance-final-snap`

```
aws rds delete-db-instance \  
  --db-instance-identifier test-instance \  
  --final-db-snapshot-identifier test-instance-final-snap
```

Output:

```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "test-instance",  
    "DBInstanceStatus": "deleting",  
    ...some output truncated...  
  }  
}
```

- Per API i dettagli, vedere [DeleteDBInstance](#) in AWS CLI Command Reference.

## delete-db-parameter-group

Il seguente esempio di codice mostra come utilizzare `delete-db-parameter-group`.

AWS CLI

Per eliminare un gruppo di parametri DB

L'esempio seguente elimina un gruppo di parametri DB.

```
aws rds delete-db-parameter-group \  
  --db-parameter-group-name mydbparametergroup
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Working with DB Parameter Groups](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DeleteDBParameter Group](#) in AWS CLI Command Reference.

## delete-db-proxy-endpoint

Il seguente esempio di codice mostra come utilizzare `delete-db-proxy-endpoint`.

## AWS CLI

Per eliminare un endpoint proxy DB per un RDS database

L'delete-db-proxy-endpointesempio seguente elimina un endpoint proxy DB per il database di destinazione.

```
aws rds delete-db-proxy-endpoint \  
  --db-proxy-endpoint-name proxyEP1
```

Output:

```
{  
  "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  
    }  
}
```

Per ulteriori informazioni, consulta [Eliminazione di un endpoint proxy](#) nella Amazon RDS User Guide e [Eliminazione di un endpoint proxy](#) nella Guida per l'utente di Amazon Aurora.

- Per i API dettagli, consulta Command Reference. [DeleteDbProxyEndpoint](#)AWS CLI

## delete-db-proxy

Il seguente esempio di codice mostra come utilizzare `delete-db-proxy`.

### AWS CLI

Per eliminare un proxy DB per un RDS database

L'esempio seguente elimina un proxy DB.

```
aws rds delete-db-proxy \  
  --db-proxy-name proxyExample
```

Output:

```
{  
  "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",  
    "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"  
  }  
}
```

Per ulteriori informazioni, consulta [Eliminazione di un RDS proxy](#) nella Amazon RDS User Guide e [Eliminazione di un RDS proxy](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [DeleteDbProxy](#) Reference AWS CLI .

## delete-db-security-group

Il seguente esempio di codice mostra come utilizzare `delete-db-security-group`.

### AWS CLI

Per eliminare un gruppo di sicurezza del DB

L'esempio seguente elimina un gruppo di sicurezza DB denominato `mysecuritygroup`.

```
aws rds delete-db-security-group \  
  --db-security-group-name mysecuritygroup
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Working with DB security groups \(piattaforma EC2 -Classic\)](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DeleteDbSecurityGroup AWS CLI](#) Command Reference.

## delete-db-shard-group

Il seguente esempio di codice mostra come utilizzare `delete-db-shard-group`.

### AWS CLI

Esempio 1: eliminare un gruppo di shard DB senza successo

L'esempio seguente mostra l'errore che si verifica quando si tenta di eliminare un gruppo di shard DB prima di eliminare tutti i database e gli schemi.

```
aws rds delete-db-shard-group \  
  --db-shard-group-identifier limitless-test-shard-grp
```

Output:

```
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.
```

Esempio 2: Per eliminare correttamente un gruppo di shard DB

L'`delete-db-shard-group`esempio seguente elimina un gruppo di shard DB dopo aver eliminato tutti i database e gli schemi, incluso lo schema. `public`

```
aws rds delete-db-shard-group \  
  --db-shard-group-identifier limitless-test-shard-grp
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Eliminazione di cluster e istanze DB Aurora](#) nella Guida per l'utente di Amazon Aurora.

- Per i API dettagli, consulta Command Reference. [DeleteDbShardGroup](#)AWS CLI

## `delete-db-snapshot`

Il seguente esempio di codice mostra come utilizzare `delete-db-snapshot`.

## AWS CLI

Per eliminare un'istantanea del DB

L'`delete-db-snapshot` seguente elimina lo snapshot DB specificato.

```
aws rds delete-db-snapshot \  
--db-snapshot-identifier mydbsnapshot
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Eliminazione di uno snapshot](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteDbSnapshot](#) Reference.



## delete-db-subnet-group

Il seguente esempio di codice mostra come utilizzare `delete-db-subnet-group`.

### AWS CLI

Per eliminare un gruppo di sottoreti DB

L'`delete-db-subnet-group`esempio seguente elimina il gruppo di sottoreti DB chiamato `mysubnetgroup`

```
aws rds delete-db-subnet-group --db-subnet-group-name mysubnetgroup
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Working with a DB Instance in a VPC](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DeleteDbSubnetGroup AWS CLI Command Reference](#).

## delete-event-subscription

Il seguente esempio di codice mostra come utilizzare `delete-event-subscription`.

### AWS CLI

Per eliminare una sottoscrizione a un evento

L'`delete-event-subscription`esempio seguente elimina la sottoscrizione all'evento specificata.

```
aws rds delete-event-subscription --subscription-name my-instance-events
```

Output:

```
{
  "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"
}
}

```

- Per API i dettagli, vedere [DeleteEventSubscription](#) in AWS CLI Command Reference.

## delete-global-cluster

Il seguente esempio di codice mostra come utilizzare `delete-global-cluster`.

### AWS CLI

Per eliminare un cluster DB globale

L'`delete-global-cluster` esempio seguente elimina un cluster DB globale compatibile con Aurora SQL My. L'output mostra il cluster che stai eliminando, ma `describe-global-clusters` i comandi successivi non elencano quel cluster DB.

```

aws rds delete-global-cluster \
  --global-cluster-identifier myglobalcluster

```

Output:

```

{
  "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": []  
  }  
}
```

Per ulteriori informazioni, consulta [Eliminazione di un database globale Aurora](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [DeleteGlobalCluster](#) Reference AWS CLI .

## **delete-option-group**

Il seguente esempio di codice mostra come utilizzare `delete-option-group`.

### AWS CLI

Per eliminare un gruppo di opzioni

L'`delete-option-group` esempio seguente elimina il gruppo di opzioni specificato.

```
aws rds delete-option-group \  
  --option-group-name myoptiongroup
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di un gruppo di opzioni](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DeleteOptionGroup AWS CLI](#) Command Reference.

## **deregister-db-proxy-targets**

Il seguente esempio di codice mostra come utilizzare `deregister-db-proxy-targets`.

### AWS CLI

Per annullare la registrazione di una destinazione proxy DB dal gruppo di destinazione del database

L'`deregister-db-proxy-targets` esempio seguente rimuove l'associazione tra il proxy `Example` e la sua destinazione.

```
aws rds deregister-db-proxy-targets \  
  --db-proxy-name Example \  
  --db-proxy-target-id ExampleTargetID
```

```
--db-proxy-name proxyExample \  
--db-instance-identifiers database-1
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di un RDS proxy](#) nella Amazon RDS User Guide e [Eliminazione di un RDS proxy](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [DeregisterDbProxyTargets](#)Reference AWS CLI .

## describe-account-attributes

Il seguente esempio di codice mostra come utilizzare `describe-account-attributes`.

### AWS CLI

Per descrivere gli attributi dell'account

L'`describe-account-attributes`esempio seguente recupera gli attributi per l' AWS account corrente.

```
aws rds describe-account-attributes
```

Output:

```
{  
  "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"
    }
  ]
}

```

- Per API i dettagli, vedere [DescribeAccountAttributes](#) in AWS CLI Command Reference.

## describe-blue-green-deployments

Il seguente esempio di codice mostra come utilizzare `describe-blue-green-deployments`.

### AWS CLI

Esempio 1: descrivere una distribuzione blu/verde di un'istanza RDS DB dopo il completamento della creazione

L'`describe-blue-green-deployments` esempio seguente recupera i dettagli di una distribuzione blu/verde dopo il completamento della creazione.

```

aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifier bgd-v53303651eexfake

```

Output:

```

{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione di una distribuzione blu/verde](#) nella Amazon RDS User Guide.

Esempio 2: Per descrivere una distribuzione blu/verde per un cluster Aurora My DB SQL

L'`describe-blue-green-deployment`esempio seguente recupera i dettagli di una distribuzione blu/verde.

```

aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake

```

Output:

```

{
  "BlueGreenDeployments": [
    {
      "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-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"
}
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione di una distribuzione blu/verde](#) nella Guida per l'utente di Amazon Aurora.

Esempio 3: Per descrivere una distribuzione blu/verde per un cluster Aurora My dopo lo switchover SQL

L'`describe-blue-green-deployment` seguente recupera i dettagli su una distribuzione blu/verde dopo che l'ambiente verde è stato promosso a ambiente di produzione.

```

aws rds describe-blue-green-deployments \
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake

```

Output:

```

{
  "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"
}
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione di una distribuzione blu/verde](#) nella Guida per l'utente di Amazon Aurora.

Esempio 4: Per descrivere una distribuzione combinata blu/verde

L'operazione `describe-blue-green-deployments` seguente recupera i dettagli di una distribuzione combinata blu/verde.

```
aws rds describe-blue-green-deployments
```

Output:

```

{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione di una distribuzione blu/verde](#) nella Amazon RDS User Guide e [Visualizzazione di una distribuzione blu/verde](#) nella Guida [per l'utente](#) di Amazon Aurora.

- Per i API dettagli, consulta Command Reference. [DescribeBlueGreenDeployments](#) AWS CLI

## describe-certificates

Il seguente esempio di codice mostra come utilizzare `describe-certificates`.

## AWS CLI

Per descrivere i certificati

L'`describe-certificates` seguente recupera i dettagli del certificato associato all'area predefinita dell'utente.

```
aws rds describe-certificates
```

Output:

```
{
  "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": "d40ddb29e3750dffa671c3140bbf5f478d1c8096",
    "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"
}

```

Per ulteriori informazioni, consulta [TLSUsareSSL/per crittografare una connessione a un'istanza DB](#) nella Amazon RDS User Guide e [UsingSSL/TLSper crittografare una connessione a un cluster DB](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta Command [DescribeCertificates](#)Reference AWS CLI .

## describe-db-cluster-backtracks

Il seguente esempio di codice mostra come utilizzare `describe-db-cluster-backtracks`.

### AWS CLI

Per descrivere i backtrack per un cluster DB

L'`describe-db-cluster-backtracks`esempio seguente recupera i dettagli sul cluster DB specificato.

```

aws rds describe-db-cluster-backtracks \
  --db-cluster-identifier mydbcluster

```

Output:

```

{
  "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"
}
]
```

Per ulteriori informazioni, consulta [Backtracking an Aurora DB cluster nella Amazon Aurora User Guide](#).

- Per API i dettagli, consulta Command [DescribeDbClusterBacktracks](#)Reference AWS CLI .

## describe-db-cluster-endpoints

Il seguente esempio di codice mostra come utilizzare `describe-db-cluster-endpoints`.

### AWS CLI

Esempio 1: Per descrivere gli endpoint del cluster DB

L'`describe-db-cluster-endpoints` esempio seguente recupera i dettagli degli endpoint del cluster DB. I tipi più comuni di cluster Aurora hanno due endpoint. Un endpoint ha un tipo. WRITER È possibile utilizzare questo endpoint per tutte le SQL istruzioni. L'altro endpoint ha un tipo. READER È possibile utilizzare questo endpoint solo per altre istruzioni SELECT di sola letturaSQL.

```
aws rds describe-db-cluster-endpoints
```

Output:

```
{
  "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-cnpxample.us-east-1.rds.amazonaws.com",
      "Status": "available",
      "EndpointType": "WRITER"
    },
    {
      "DBClusterIdentifier": "mydbcluster",
      "Endpoint": "mydbcluster.cluster-ro-cnpxample.us-
east-1.rds.amazonaws.com",
      "Status": "available",
      "EndpointType": "READER"
    }
  ]
}

```

Esempio 2: Per descrivere gli endpoint del cluster DB di un singolo cluster DB

L'operazione `aws rds describe-db-cluster-endpoints` seguente recupera i dettagli per gli endpoint del cluster DB di un singolo cluster DB specificato. I cluster Aurora Serverless hanno un solo endpoint con un tipo di `WRITER`

```

aws rds describe-db-cluster-endpoints \
  --db-cluster-identifier serverless-cluster

```

Output:

```

{
  "DBClusterEndpoints": [
    {
      "Status": "available",
      "Endpoint": "serverless-cluster.cluster-cnpxample.us-
east-1.rds.amazonaws.com",
      "DBClusterIdentifier": "serverless-cluster",
      "EndpointType": "WRITER"
    }
  ]
}

```

```
    }
  ]
}
```

Per ulteriori informazioni, consulta [Amazon Aurora Connection Management](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeDbClusterEndpoints](#) Reference.

## describe-db-cluster-parameter-groups

Il seguente esempio di codice mostra come utilizzare `describe-db-cluster-parameter-groups`.

### AWS CLI

Per descrivere i gruppi di parametri del cluster DB

L'esempio seguente recupera i dettagli per i gruppi di parametri del cluster DB.

```
aws rds describe-db-cluster-parameter-groups
```

Output:

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Working with DB Parameter Groups e DB Cluster Parameter Groups](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta [DescribeDbClusterParameterGroups AWS CLI Command Reference](#).

## describe-db-cluster-parameters

Il seguente esempio di codice mostra come utilizzare `describe-db-cluster-parameters`.

### AWS CLI

Esempio 1: Per descrivere i parametri in un gruppo di parametri del cluster DB

L'`describe-db-cluster-parameters` seguente recupera i dettagli sui parametri in un gruppo di parametri del cluster DB.

```

aws rds describe-db-cluster-parameters \
  --db-cluster-parameter-group-name mydbclusterpg

```

Output:

```

{
  "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...
  ]
}

```

Esempio 2: Elencare solo i nomi dei parametri in un gruppo di parametri del cluster DB

L'`describe-db-cluster-parameters` esempio seguente recupera solo i nomi dei parametri in un gruppo di parametri del cluster DB.

```

aws rds describe-db-cluster-parameters \
  --db-cluster-parameter-group-name default.aurora-mysql5.7 \
  --query 'Parameters[].[ParameterName:ParameterName]'

```

## Output:

```
[
  {
    "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...
]
```

Esempio 3: Per descrivere solo i parametri modificabili in un gruppo di parametri del cluster DB

L'execute-db-cluster-parametersesempio seguente recupera solo i nomi dei parametri che è possibile modificare in un gruppo di parametri del cluster DB.

```
aws rds describe-db-cluster-parameters \
  --db-cluster-parameter-group-name default.aurora-mysql5.7 \
  --query 'Parameters[].[ParameterName:ParameterName,IsModifiable:IsModifiable] |
[?IsModifiable == `true`]'
```

## Output:

```
[
  {
    "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...
  ]
]

```

Esempio 4: Per descrivere solo i parametri booleani modificabili in un gruppo di parametri del cluster DB

L'output di `aws rds describe-db-cluster-parameters` seguente recupera solo i nomi dei parametri che è possibile modificare in un gruppo di parametri del cluster DB e che hanno un tipo di dati booleano.

```

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`]'

```

Output:

```

[
  {
    "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...
}
]

```

Per ulteriori informazioni, consulta [Working with DB Parameter Groups e DB Cluster Parameter Groups](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta [DescribeDbClusterParameters AWS CLI Command Reference](#).

## describe-db-cluster-snapshot-attributes

Il seguente esempio di codice mostra come utilizzare `describe-db-cluster-snapshot-attributes`.

### AWS CLI

Per descrivere i nomi e i valori degli attributi per uno snapshot del cluster DB

L'esempio seguente recupera i dettagli dei nomi e dei valori degli attributi per lo snapshot del cluster DB specificato.

```

aws rds describe-db-cluster-snapshot-attributes \
  --db-cluster-snapshot-identifier myclustersnapshot

```

Output:

```

{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotIdentifier": "myclustersnapshot",
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123456789012"
        ]
      }
    ]
  }
}

```

```

    }
  ]
}

```

Per ulteriori informazioni, consulta [Condivisione di uno snapshot del cluster DB](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta AWS CLI Command [DescribeDbClusterSnapshotAttributesReference](#).

## describe-db-cluster-snapshots

Il seguente esempio di codice mostra come utilizzare `describe-db-cluster-snapshots`.

### AWS CLI

Per descrivere uno snapshot del cluster DB per un cluster DB

L'`describe-db-cluster-snapshots` seguente recupera i dettagli per le istantanee del cluster DB per il cluster DB specificato.

```
aws rds describe-db-cluster-snapshots \
  --db-cluster-identifier mydbcluster
```

Output:

```
{
  "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
  }
]
}

```

Per ulteriori informazioni, consulta [Creazione di uno snapshot del cluster DB](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta AWS CLI Command [DescribeDbClusterSnapshots](#)Reference.

## describe-db-clusters

Il seguente esempio di codice mostra come utilizzare `describe-db-clusters`.

### AWS CLI

Esempio 1: Per descrivere un cluster DB

L'`describe-db-clusters`esempio seguente recupera i dettagli del cluster DB specificato.

```
aws rds describe-db-clusters \
  --db-cluster-identifier mydbcluster
```

Output:

```
{
  "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-cnpeexample.us-
east-1.rds.amazonaws.com",
      "ReaderEndpoint": "mydbcluster.cluster-ro-cnpeexample.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
  }
]
}

```

Esempio 2: per elencare determinati attributi di tutti i cluster DB

L'`describe-db-clusters` esempio seguente recupera solo gli `ReaderEndpoint` attributi `DBClusterIdentifierEndpoint`, e di tutti i cluster DB nella regione corrente. AWS

```

aws rds describe-db-clusters \
  --query 'DBClusters[].
{DBClusterIdentifier:DBClusterIdentifier,Endpoint:Endpoint,ReaderEndpoint:ReaderEndpoint}'

```

Output:

```

[
  {
    "Endpoint": "cluster-57-2020-05-01-2270.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
    "ReaderEndpoint": "cluster-57-2020-05-01-2270.cluster-ro-cnpexample.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-cnpxexample.us-
east-1.rds.amazonaws.com",
    "DBClusterIdentifier": "cluster-57-2020-05-01-4615"
  },
  {
    "Endpoint": "pg2-cluster.cluster-cnpxexample.us-east-1.rds.amazonaws.com",
    "ReaderEndpoint": "pg2-cluster.cluster-ro-cnpxexample.us-
east-1.rds.amazonaws.com",
    "DBClusterIdentifier": "pg2-cluster"
  },
  ...output omitted...
}
]

```

Esempio 3: elencare i cluster DB con un attributo specifico

L'example seguente recupera solo gli Engine attributi DBClusterIdentifier e dei cluster DB che utilizzano il aurora-postgresql motore DB.

```

aws rds describe-db-clusters \
  --query 'DBClusters[].[DBClusterIdentifier:DBClusterIdentifier,Engine:Engine] |
  [?Engine == `aurora-postgresql`]'

```

Output:

```

[
  {
    "Engine": "aurora-postgresql",
    "DBClusterIdentifier": "pg2-cluster"
  }
]

```

Per ulteriori informazioni, consulta [Amazon Aurora DB Clusters](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [DescribeDbClusters](#)Reference AWS CLI .

## describe-db-engine-versions

Il seguente esempio di codice mostra come utilizzare `describe-db-engine-versions`.

## AWS CLI

Per descrivere le versioni del motore DB per il motore My SQL DB

L'output di `aws rds describe-db-engine-versions` seguente mostra i dettagli su ciascuna delle versioni del motore DB per il motore DB specificato.

```
aws rds describe-db-engine-versions \
  --engine mysql
```

Output:

```
{
  "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...
    }
  ]
}
```



```
]
}
```

Per ulteriori informazioni, consulta [What Is Amazon Relational Database Service \(RDSAmazon\)?](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DescribeDBEngine Versions](#) in AWS CLI Command Reference.

## describe-db-instance-automated-backups

Il seguente esempio di codice mostra come utilizzare `describe-db-instance-automated-backups`.

### AWS CLI

Per descrivere i backup automatici per un'istanza DB

L'esempio seguente mostra i dettagli sui backup automatici per l'istanza DB specificata. I dettagli includono backup automatici replicati in altre regioni. AWS

```
aws rds describe-db-instance-automated-backups \
  --db-instance-identifier new-orcl-db
```

Output:

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta la sezione [Ricerca di informazioni sui backup replicati](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeDbInstanceAutomatedBackupsReference](#).

## describe-db-instances

Il seguente esempio di codice mostra come utilizzare `describe-db-instances`.

### AWS CLI

Per descrivere un'istanza DB

L'`describe-db-instances` seguente recupera i dettagli sull'istanza DB specificata.

```

aws rds describe-db-instances \
  --db-instance-identifier mydbinstancecf

```

Output:

```

{
  "DBInstances": [
    {
      "DBInstanceIdentifier": "mydbinstancecf",
      "DBInstanceClass": "db.t3.small",
      "Engine": "mysql",
      "DBInstanceStatus": "available",

```

```

        "MasterUsername": "masterawsuser",
        "Endpoint": {
            "Address": "mydbinstancecf.abcexample.us-east-1.rds.amazonaws.com",
            "Port": 3306,
            "HostedZoneId": "Z2R2ITUGPM61AM"
        },
        ...some output truncated...
    }
]
}

```

- Per API i dettagli, vedete [DescribeDBInstances](#) in AWS CLI Command Reference.

## describe-db-log-files

Il seguente esempio di codice mostra come utilizzare `describe-db-log-files`.

### AWS CLI

Per descrivere i file di log per un'istanza DB

L'esempio seguente recupera i dettagli sui file di registro per l'istanza DB specificata.

```

aws rds describe-db-log-files -\
    -db-instance-identifier test-instance

```

Output:

```

{
  "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"
    }
]
}

```

- Per API i dettagli, vedere [DescribeDbLogFiles](#) in AWS CLI Command Reference.

## describe-db-parameter-groups

Il seguente esempio di codice mostra come utilizzare `describe-db-parameter-groups`.

### AWS CLI

Per descrivere il gruppo di parametri del database

L'esempio seguente recupera i dettagli sui gruppi di parametri DB.

```
aws rds describe-db-parameter-groups
```

Output:

```

{
  "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...
]
}

```

Per ulteriori informazioni, consulta [Working with DB Parameter Groups](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DescribeDBParameter Groups](#) in AWS CLI Command Reference.

## describe-db-parameters

Il seguente esempio di codice mostra come utilizzare `describe-db-parameters`.

### AWS CLI

Per descrivere i parametri in un gruppo di parametri DB

L'execute `aws rds describe-db-parameters` seguente recupera i dettagli del gruppo di parametri DB specificato.

```
aws rds describe-db-parameters \
  --db-parameter-group-name mydbpg
```

Output:

```
{
  "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...
  ]
}
```

Per ulteriori informazioni, consulta [Working with DB Parameter Groups](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DescribeDBParameters](#) in AWS CLI Command Reference.

## describe-db-proxies

Il seguente esempio di codice mostra come utilizzare `describe-db-proxies`.

### AWS CLI

Per descrivere un proxy DB per un RDS database

L'`describe-db-proxies` esempio seguente restituisce informazioni sui proxy DB.

```
aws rds describe-db-proxies
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Viewing an RDS Proxy](#) nella Amazon RDS User Guide e [Viewing an RDS Proxy](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta [DescribeDbProxies AWS CLI Command Reference](#).



## describe-db-proxy-endpoints

Il seguente esempio di codice mostra come utilizzare `describe-db-proxy-endpoints`.

### AWS CLI

Per descrivere gli endpoint di un proxy DB

L'`describe-db-proxy-endpoint`sempio seguente restituisce informazioni sugli endpoint proxy DB.

```
aws rds describe-db-proxy-endpoints
```

Output:

```
{
  "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
  }
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione di un endpoint proxy](#) nella Amazon RDS User Guide e [Creazione di un endpoint proxy](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [DescribeDbProxyEndpointsReference AWS CLI](#) .

## describe-db-proxy-target-groups

Il seguente esempio di codice mostra come utilizzare `describe-db-proxy-target-groups`.

### AWS CLI

Per descrivere gli endpoint di un proxy DB

L'`describe-db-proxy-target-groups`esempio seguente restituisce informazioni sui gruppi target del proxy DB.

```
aws rds describe-db-proxy-target-groups \
  --db-proxy-name proxyExample
```

Output:

```
{
  "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",
    "UpdateDate": "2023-05-02T18:41:21.762000+00:00"
  }
}

```

Per ulteriori informazioni, consulta [Viewing an RDS Proxy](#) nella Amazon RDS User Guide e [Viewing an RDS Proxy](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta [DescribeDbProxyTargetGroups AWS CLI Command Reference](#).

## describe-db-proxy-targets

Il seguente esempio di codice mostra come utilizzare `describe-db-proxy-targets`.

### AWS CLI

Per descrivere gli obiettivi del proxy DB

L'`describe-db-proxy-targets` esempio seguente restituisce informazioni sugli obiettivi proxy DB.

```
aws rds describe-db-proxy-targets \
  --db-proxy-name proxyExample
```

Output:

```

{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Visualizzazione di un RDS proxy](#) nella Amazon RDS User Guide e [Visualizzazione di un RDS proxy](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta [DescribeDbProxyTargets AWS CLI Command Reference](#).

## describe-db-recommendations

Il seguente esempio di codice mostra come utilizzare `describe-db-recommendations`.

### AWS CLI

Esempio 1: per elencare tutti i consigli del DB

L'`describe-db-recommendations` seguente elenca tutti i consigli DB presenti nel tuo AWS account.

```
aws rds describe-db-recommendations
```

Output:

```
{
  "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"
            }
        ]
    }
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione e risposta ai RDS consigli di Amazon](#) nella Guida per l'RDSutente di Amazon e [Visualizzazione e risposta ai RDS consigli di Amazon](#) nella Guida per l'utente di Amazon Aurora.

Esempio 2: per elencare i consigli di database ad alta severità

L'`describe-db-recommendation`esempio seguente elenca i consigli DB ad alta severità nel tuo AWS account.

```

aws rds describe-db-recommendations \
  --filters Name=severity,Values=high

```

Output:

```

{

```

```
"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.\n\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.\n\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/"
      }
    ]
  }
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione e risposta ai RDS consigli di Amazon](#) nella Guida per l'RDSutente di Amazon e [Visualizzazione e risposta ai RDS consigli di Amazon](#) nella Guida per l'utente di Amazon Aurora.

Esempio 3: per elencare i consigli del database per un'istanza DB specificata

L'`describe-db-recommendation`sempio seguente elenca tutti i consigli DB per un'istanza DB specificata.

```

aws rds describe-db-recommendations \
  --filters Name=dbi-resource-id,Values=database-1

```

Output:



```
{
  "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"
  }
]
}
]
}

```

Per ulteriori informazioni, consulta [Visualizzazione e risposta ai RDS consigli di Amazon](#) nella Guida per l'utente di Amazon e [Visualizzazione e risposta ai RDS consigli di Amazon](#) nella Guida per l'utente di Amazon Aurora.

Esempio 4: per elencare tutti i consigli DB attivi

L'output di `aws rds describe-db-recommendations` seguente elenca tutti i consigli DB attivi nel tuo AWS account.

```
aws rds describe-db-recommendations \
  --filters Name=status,Values=active
```

Output:

```
{
  "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"
        }
    ]
}
]
```

```
}
```

Per ulteriori informazioni, consulta [Visualizzazione e risposta ai RDS consigli di Amazon](#) nella Guida per l'utente di Amazon e [Visualizzazione e risposta ai RDS consigli di Amazon](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [DescribeDbRecommendations](#) Reference AWS CLI .

## describe-db-security-groups

Il seguente esempio di codice mostra come utilizzare `describe-db-security-groups`.

### AWS CLI

Per elencare i gruppi di sicurezza del DB

L'`describe-db-security-groups` seguente elenca i gruppi di sicurezza DB.

```
aws rds describe-db-security-groups
```

Output:

```
{
  "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"
    }
  ]
}
```

```

    }
  ]
}

```

Per ulteriori informazioni, consulta la sezione [Elenco dei gruppi di sicurezza DB disponibili](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DescribeDbSecurityGroups AWS CLI Command Reference](#).

## describe-db-shard-groups

Il seguente esempio di codice mostra come utilizzare `describe-db-shard-groups`.

### AWS CLI

Esempio 1: Per descrivere i gruppi di shard DB

L'`describe-db-shard-groups` seguente recupera i dettagli dei gruppi di shard DB.

```
aws rds describe-db-shard-groups
```

Output:

```

{
  "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-cekyceexample.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"
    }
]
}
```

Per ulteriori informazioni, consulta [Amazon Aurora DB Clusters](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [DescribeDbShardGroupsReference AWS CLI](#) .

## describe-db-snapshot-attributes

Il seguente esempio di codice mostra come utilizzare `describe-db-snapshot-attributes`.

### AWS CLI

Per descrivere i nomi e i valori degli attributi per un'istantanea del DB

L'`describe-db-snapshot-attributes`esempio seguente descrive i nomi e i valori degli attributi per uno snapshot DB.

```
aws rds describe-db-snapshot-attributes \
  --db-snapshot-identifier mydbsnapshot
```

Output:

```
{
  "DBSnapshotAttributesResult": {
    "DBSnapshotIdentifier": "mydbsnapshot",
    "DBSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123456789012",
          "210987654321"
        ]
      }
    ]
  }
}
```

```
}
```

Per ulteriori informazioni, consulta [Sharing a DB Snapshot](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DescribeDbSnapshotAttributes AWS CLI Command Reference](#).

## describe-db-snapshots

Il seguente esempio di codice mostra come utilizzare `describe-db-snapshots`.

### AWS CLI

Esempio 1: descrivere uno snapshot DB per un'istanza DB

L'`describe-db-snapshots` seguente recupera i dettagli di uno snapshot DB per un'istanza DB.

```
aws rds describe-db-snapshots \  
  --db-snapshot-identifier mydbsnapshot
```

Output:

```
{  
  "DBSnapshots": [  
    {  
      "DBSnapshotIdentifier": "mydbsnapshot",  
      "DBInstanceIdentifier": "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Creating a DB Snapshot](#) nella Amazon RDS User Guide.

Esempio 2: per trovare il numero di istantanee scattate manualmente

L'`describe-db-snapshots` seguente utilizza l'operatore `length` nell'opzione `--query` per restituire il numero di istantanee manuali che sono state scattate in una particolare AWS regione.

```

aws rds describe-db-snapshots \
  --snapshot-type manual \
  --query "length(*[].[DBSnapshots:SnapshotType])" \
  --region eu-central-1

```

Output:

```
35
```

Per ulteriori informazioni, consulta [Creating a DB Snapshot](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DescribeDBSnapshots](#) in AWS CLI Command Reference.

## **describe-db-subnet-groups**

Il seguente esempio di codice mostra come utilizzare `describe-db-subnet-groups`.

### AWS CLI

Per descrivere un gruppo di sottoreti DB

L'`describe-db-subnet-groups` seguente recupera i dettagli del gruppo di sottoreti DB specificato.

**aws rds describe-db-subnet-groups**

## Output:

```
{
  "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"
    }
  ]
}
```

```

    }
  ]
}

```

Per ulteriori informazioni, consulta [Amazon Virtual Private Cloud VPCs e Amazon RDS](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DescribeDbSubnetGroups AWS CLI](#) Command Reference.

## describe-engine-default-cluster-parameters

Il seguente esempio di codice mostra come utilizzare `describe-engine-default-cluster-parameters`.

### AWS CLI

Per descrivere le informazioni predefinite sul motore e sui parametri di sistema per il motore di database Aurora

L'`describe-engine-default-cluster-parameters` esempio seguente recupera i dettagli del motore predefinito e delle informazioni sui parametri di sistema per i cluster Aurora DB con SQL compatibilità My 5.7.

```

aws rds describe-engine-default-cluster-parameters \
  --db-parameter-group-family aurora-mysql5.7

```

Output:

```

{
  "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...
  ]
}
}

```

Per ulteriori informazioni, consulta [Working with DB Parameter Groups e DB Cluster Parameter Groups](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta [DescribeEngineDefaultClusterParameters AWS CLI Command Reference](#).

## describe-engine-default-parameters

Il seguente esempio di codice mostra come utilizzare `describe-engine-default-parameters`.

### AWS CLI

Per descrivere il motore predefinito e le informazioni sui parametri di sistema per il motore di database

L'`describe-engine-default-parameters` seguente recupera i dettagli per il motore predefinito e le informazioni sui parametri di sistema per le istanze My SQL 5.7 DB.

```
aws rds describe-engine-default-parameters \
  --db-parameter-group-family mysql5.7
```

Output:

```
{
  "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...
    ]
}
}
```

Per ulteriori informazioni, consulta [Working with DB Parameter Groups](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DescribeEngineDefaultParameters AWS CLI Command Reference](#).

## describe-event-categories

Il seguente esempio di codice mostra come utilizzare `describe-event-categories`.

### AWS CLI

Per descrivere le categorie di eventi

L'esempio seguente recupera i dettagli sulle categorie di eventi per tutte le fonti di eventi disponibili.

```
aws rds describe-event-categories
```

Output:

```
{
  "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"
    ]
}
]
```

- Per API i dettagli, vedere [DescribeEventCategories](#) in AWS CLI Command Reference.

## describe-event-subscriptions

Il seguente esempio di codice mostra come utilizzare `describe-event-subscriptions`.

### AWS CLI

Per descrivere le sottoscrizioni agli eventi

Questo esempio descrive tutti gli abbonamenti agli RDS eventi Amazon per l' AWS account corrente.

```
aws rds describe-event-subscriptions
```

Output:

```
{
  "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...
  ]
}
```

- Per API i dettagli, consulta [DescribeEventSubscriptions AWS CLI Command Reference](#).

## describe-events

Il seguente esempio di codice mostra come utilizzare `describe-events`.

## AWS CLI

Per descrivere gli eventi

L' `describe-events` esempio seguente recupera i dettagli degli eventi che si sono verificati per l'istanza DB specificata.

```
aws rds describe-events \  
  --source-identifier test-instance \  
  --source-type db-instance
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [DescribeEvents](#) in AWS CLI Command Reference.



## describe-export-tasks

Il seguente esempio di codice mostra come utilizzare `describe-export-tasks`.

### AWS CLI

Per descrivere le attività di esportazione delle istantanee

L'`describe-export-tasks` seguente restituisce informazioni sulle esportazioni di snapshot in Amazon S3.

```
aws rds describe-export-tasks
```

Output:

```
{
  "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
    }
]
}
```

Per ulteriori informazioni, consulta [Monitoring Snapshot Exports](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DescribeExportTasks AWS CLI Command Reference](#).

## describe-global-clusters

Il seguente esempio di codice mostra come utilizzare `describe-global-clusters`.

### AWS CLI

Per descrivere i cluster DB globali

L'esempio seguente elenca i cluster DB globali Aurora nella regione corrente. AWS

```
aws rds describe-global-clusters
```

Output:

```
{
  "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": []
    }
  ]
}
```

Per ulteriori informazioni, consulta [Gestire un database globale Aurora](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta AWS CLI Command [DescribeGlobalClusters](#)Reference.

## describe-option-group-options

Il seguente esempio di codice mostra come utilizzare `describe-option-group-options`.

### AWS CLI

Per descrivere tutte le opzioni disponibili

L'`describe-option-group-options`esempio seguente elenca due opzioni per un'istanza di Oracle Database 19c.

```
aws rds describe-option-group-options \  
  --engine-name oracle-ee \  
  --major-engine-version 19 \  
  --max-items 2
```

Output:

```
{  
  "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=="
}

```

Per ulteriori informazioni, consulta [Elencare le opzioni e le impostazioni delle opzioni per un gruppo di opzioni](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DescribeOptionGroupOptions AWS CLI](#) Command Reference.

## describe-option-groups

Il seguente esempio di codice mostra come utilizzare `describe-option-groups`.

### AWS CLI

Per descrivere i gruppi di opzioni disponibili

L'`describe-option-groups` seguente elenca i gruppi di opzioni per un'istanza di Oracle Database 19c.

```
aws rds describe-option-groups \  
  --engine-name oracle-ee \  
  --major-engine-version 19
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Elencare le opzioni e le impostazioni delle opzioni per un gruppo di opzioni](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DescribeOptionGroups AWS CLI](#) Command Reference.

## `describe-orderable-db-instance-options`

Il seguente esempio di codice mostra come utilizzare `describe-orderable-db-instance-options`.

AWS CLI

Per descrivere le opzioni delle istanze DB ordinabili

L'`describe-orderable-db-instance-options` seguente recupera i dettagli sulle opzioni ordinabili per le istanze DB che eseguono il motore My DB. SQL

```
aws rds describe-orderable-db-instance-options \  
  --engine-name mysql5.7 \  
  --major-engine-version 5.7
```

```
--engine mysql
```

Output:

```
{
  "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...
}
```

- Per API i dettagli, vedere [DescribeOrderableDBInstanceOptions](#) in AWS CLI Command Reference.

## describe-pending-maintenance-actions

Il seguente esempio di codice mostra come utilizzare `describe-pending-maintenance-actions`.

### AWS CLI

Per elencare le risorse con almeno un'azione di manutenzione in sospeso

L'`describe-pending-maintenance-actions` esempio seguente elenca l'azione di manutenzione in sospeso per un'istanza DB.

```
aws rds describe-pending-maintenance-actions
```

Output:

```
{
  "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"
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, consulta [la sezione Manutenzione di un'istanza DB](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [DescribePendingMaintenanceActions AWS CLI Command Reference](#).

## describe-reserved-db-instances-offerings

Il seguente esempio di codice mostra come utilizzare `describe-reserved-db-instances-offerings`.

### AWS CLI

Per descrivere le offerte di istanze DB riservate

L'esempio seguente recupera i dettagli sulle opzioni di istanza DB riservate per Oracle

```
aws rds describe-reserved-db-instances-offerings \
  --product-description oracle
```

Output:

```
{
  "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...
  ]
}
```

- Per API i dettagli, vedere [DescribeReservedDbInstancesOfferings](#) in AWS CLI Command Reference.



## describe-reserved-db-instances

Il seguente esempio di codice mostra come utilizzare `describe-reserved-db-instances`.

### AWS CLI

Per descrivere le istanze DB riservate

L'`describe-reserved-db-instances` esempio seguente recupera i dettagli su tutte le istanze DB riservate nell'account corrente. AWS

```
aws rds describe-reserved-db-instances
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione [Istanze database riservate per Amazon RDS nella Amazon RDS User Guide](#).

- Per API i dettagli, consulta [DescribeReservedDbInstances AWS CLI Command Reference](#).

## describe-source-regions

Il seguente esempio di codice mostra come utilizzare `describe-source-regions`.

### AWS CLI

Per descrivere le regioni di origine

L'esempio seguente recupera i dettagli su tutte le AWS regioni di origine. Mostra inoltre che i backup automatici possono essere replicati solo dagli Stati Uniti occidentali (Oregon) alla AWS regione di destinazione, Stati Uniti orientali (Virginia settentrionale).

```
aws rds describe-source-regions \  
  --region us-east-1
```

Output:

```
{  
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione [Ricerca di informazioni sui backup replicati](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeSourceRegions](#) Reference.

## describe-valid-db-instance-modifications

Il seguente esempio di codice mostra come utilizzare `describe-valid-db-instance-modifications`.

### AWS CLI

Per descrivere le modifiche valide per un'istanza DB

L'`describe-valid-db-instance-modifications`esempio seguente recupera i dettagli sulle modifiche valide per l'istanza DB specificata.

```
aws rds describe-valid-db-instance-modifications \
  --db-instance-identifier test-instance
```

Output:

```
{
  "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"
}
]
}
```

- Per API i dettagli, vedere [DescribeValidDbInstanceModifications](#) in AWS CLI Command Reference.

## download-db-log-file-portion

Il seguente esempio di codice mostra come utilizzare `download-db-log-file-portion`.

### AWS CLI

Per scaricare un file di registro del DB

L'esempio seguente scarica solo la parte più recente del file di registro, salvandola in un file locale denominato `tail.txt`.

```
aws rds download-db-log-file-portion \  
  --db-instance-identifier test-instance \  
  --log-file-name log.txt \  
  --output text > tail.txt
```

Per scaricare l'intero file, è necessario includere il `--starting-token 0` parametro. L'esempio seguente salva l'output in un file locale denominato `full.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
```

Il file salvato potrebbe contenere righe vuote. Appaiono alla fine di ogni parte del file di registro durante il download. Questo in genere non causa alcun problema nell'analisi del file di registro.

- Per API i dettagli, vedere [DownloadDbLogFilePortion](#) in AWS CLI Command Reference.

## generate-auth-token

Il seguente esempio di codice mostra come utilizzare `generate-auth-token`.

### AWS CLI

Per generare un token di autenticazione



L'generate-db-auth-token esempio seguente genera un token di autenticazione da utilizzare con l'autenticazione IAM del database.

```
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
```

Output:

```
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
```

- Per API i dettagli, vedere [GenerateAuthToken](#) in AWS CLI Command Reference.

## generate-db-auth-token

Il seguente esempio di codice mostra come utilizzare generate-db-auth-token.

### AWS CLI

Per generare un token IAM di autenticazione

L'generate-db-auth-token esempio seguente genera un token di IAM autenticazione per connettersi a un database.

```
aws rds generate-db-auth-token \  
  --hostname mydb.123456789012.us-east-1.rds.amazonaws.com \  
  --port 3306 \  
  --region us-east-1 \  
  --username db_user
```

Output:

```
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
```

Per ulteriori informazioni, consulta [Connessione all'istanza DB utilizzando IAM l'autenticazione](#) nella Amazon RDS User Guide e [Connessione al cluster DB tramite IAM l'autenticazione](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta [GenerateDbAuthToken AWS CLI](#) Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag su una RDS risorsa Amazon

L'`list-tags-for-resource` esempio seguente elenca tutti i tag su un'istanza DB.

```
aws rds list-tags-for-resource \
  --resource-name arn:aws:rds:us-east-1:123456789012:db:orcl1
```

Output:

```
{
  "TagList": [
    {
      "Key": "Environment",
      "Value": "test"
    },
    {
      "Key": "Name",
      "Value": "MyDatabase"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Tagging Amazon RDS Resources](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI](#) Command Reference.

## modify-certificates

Il seguente esempio di codice mostra come utilizzare `modify-certificates`.

### AWS CLI

Per sovrascrivere temporaneamente il TLS certificato `system-defaultSSL`/per le nuove istanze DB

L'`modify-certificates`esempio seguente sovrascrive temporaneamente il `system-default/certificate` per le nuove istanze DBSSL. TLS

```
aws rds modify-certificates \  
  --certificate-identifier rds-ca-2019
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Rotating yourSSL/TLSertificate](#) nella Amazon RDS User Guide e [Rotating yourSSL/TLSertificate](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta Command [ModifyCertificates](#)Reference AWS CLI .

## modify-current-db-cluster-capacity

Il seguente esempio di codice mostra come utilizzare `modify-current-db-cluster-capacity`.

### AWS CLI

Per scalare la capacità di un cluster DB Aurora Serverless

L'`modify-current-db-cluster-capacity` esempio seguente ridimensiona la capacità di un cluster DB Aurora Serverless a 8.

```
aws rds modify-current-db-cluster-capacity \  
  --db-cluster-identifier mydbcluster \  
  --capacity 8
```

Output:

```
{  
  "DBClusterIdentifier": "mydbcluster",  
  "PendingCapacity": 8,  
  "CurrentCapacity": 1,  
  "SecondsBeforeTimeout": 300,  
  "TimeoutAction": "ForceApplyCapacityChange"  
}
```

Per ulteriori informazioni, consulta [Scalare manualmente la capacità del cluster DB Aurora Serverless v1](#) nella Amazon Aurora User Guide.

- Per i API dettagli, consulta Command Reference. [ModifyCurrentDbClusterCapacity](#) AWS CLI

## `modify-db-cluster-endpoint`

Il seguente esempio di codice mostra come utilizzare `modify-db-cluster-endpoint`.

AWS CLI

Per modificare un endpoint del cluster DB personalizzato

L'`modify-db-cluster-endpoint` esempio seguente modifica l'endpoint personalizzato del cluster DB specificato.

```
aws rds modify-db-cluster-endpoint \  
  --db-cluster-endpoint-identifier mycustomendpoint \  
  --static-members dbinstance1 dbinstance2 dbinstance3
```

Output:

```
{  
  "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"
}

```

Per ulteriori informazioni, consulta [Amazon Aurora Connection Management](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta AWS CLI Command [ModifyDbClusterEndpoint](#) Reference.

## modify-db-cluster-parameter-group

Il seguente esempio di codice mostra come utilizzare `modify-db-cluster-parameter-group`.

### AWS CLI

Per modificare i parametri in un gruppo di parametri del cluster DB

L'`modify-db-cluster-parameter-group` esempio seguente modifica i valori dei parametri in un gruppo di parametri del cluster DB.

```

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"

```

Output:

```
{
  "DBClusterParameterGroupName": "mydbclusterpg"
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di parametri DB e i gruppi di parametri del cluster DB](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta [ModifyDbClusterParameterGroup AWS CLI Command Reference](#).

## modify-db-cluster-snapshot-attribute

Il seguente esempio di codice mostra come utilizzare `modify-db-cluster-snapshot-attribute`.

### AWS CLI

Per modificare un attributo di snapshot del cluster DB

L'esempio seguente `modify-db-cluster-snapshot-attribute` apporta modifiche all'attributo di snapshot del cluster DB specificato.

```
aws rds modify-db-cluster-snapshot-attribute \
  --db-cluster-snapshot-identifier myclustersnapshot \
  --attribute-name restore \
  --values-to-add 123456789012
```

Output:

```
{
  "DBClusterSnapshotAttributesResult": {
    "DBClusterSnapshotIdentifier": "myclustersnapshot",
    "DBClusterSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "123456789012"
        ]
      }
    ]
  }
}
```

Per ulteriori informazioni, consulta [Restoring from a DB Cluster Snapshot](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta Command [ModifyDbClusterSnapshotAttribute](#) Reference AWS CLI .

## modify-db-cluster

Il seguente esempio di codice mostra come utilizzare `modify-db-cluster`.

### AWS CLI

Esempio 1: modificare un cluster DB

L'esempio seguente modifica la password dell'utente principale per il cluster DB denominato `cluster-2` e imposta il periodo di conservazione del backup su 14 giorni. Il `--apply-immediately` parametro fa sì che le modifiche vengano apportate immediatamente, anziché attendere la finestra di manutenzione successiva.

```
aws rds modify-db-cluster \  
  --db-cluster-identifier cluster-2 \  
  --backup-retention-period 14 \  
  --master-user-password newpassword99 \  
  --apply-immediately
```

Output:

```
{  
  "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": []
  }
}
```



Per ulteriori informazioni, consulta [Modificare un cluster Amazon Aurora DB](#) nella Guida per l'utente di Amazon Aurora.

Esempio 2: associare un gruppo VPC di sicurezza a un cluster DB

L'`modify-db-instance` seguente associa un gruppo di VPC sicurezza specifico e rimuove i gruppi di sicurezza DB da un cluster DB.

```
aws rds modify-db-cluster \  
  --db-cluster-identifier dbName \  
  --vpc-security-group-ids sg-ID
```

Output:

```
{  
  "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...
}
}

```

Per ulteriori informazioni, consulta [Controllare l'accesso con gruppi di sicurezza](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta [ModifyDbCluster AWS CLI Command Reference](#).

## modify-db-instance

Il seguente esempio di codice mostra come utilizzare `modify-db-instance`.

### AWS CLI

Esempio 1: modificare un'istanza DB

L'`modify-db-instance` esempio seguente associa un gruppo di opzioni e un gruppo di parametri a un'istanza DB di Microsoft SQL Server compatibile. Il `--apply-immediately` parametro fa sì che l'opzione e i gruppi di parametri vengano associati immediatamente, anziché attendere la finestra di manutenzione successiva.

```

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

```

Output:

```
{
```

```
"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
}
}
```

Per ulteriori informazioni, consulta [Modifying an Amazon RDS DB Instance](#) nella Amazon RDS User Guide.

Esempio 2: associare un gruppo VPC di sicurezza a un'istanza DB

L'`modify-db-instance` seguente associa un gruppo di VPC sicurezza specifico e rimuove i gruppi di sicurezza DB da un'istanza DB:

```
aws rds modify-db-instance \  
  --db-instance-identifier dbName \  
  --vpc-security-group-ids sg-ID
```

Output:

```
{  
  "DBInstance": {  
    "DBInstanceIdentifier": "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,  
    "ReadReplicaDBInstanceIdentifiers": [],  
    "LicenseModel": "general-public-license",
```

```
    ... output ommited ...  
  }  
}
```

Per ulteriori informazioni, consulta [Controllare l'accesso con gruppi di sicurezza](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [ModifyDBInstance](#) in AWS CLI Command Reference.

## modify-db-parameter-group

Il seguente esempio di codice mostra come utilizzare `modify-db-parameter-group`.

### AWS CLI

Per modificare un gruppo di parametri DB

L'esempio seguente modifica il valore del `clr_enabled` parametro in un gruppo di parametri DB. Il `--apply-immediately` parametro fa sì che il gruppo di parametri DB venga modificato immediatamente, anziché attendere la finestra di manutenzione successiva.

```
aws rds modify-db-parameter-group \  
  --db-parameter-group-name test-sqlserver-se-2017 \  
  --parameters "ParameterName='clr_enabled',ParameterValue=1,ApplyMethod=immediate"
```

Output:

```
{  
  "DBParameterGroupName": "test-sqlserver-se-2017"  
}
```

Per ulteriori informazioni, consulta [Modifying Parameters in a DB Parameter Group](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [ModifyDBParameter Group](#) in AWS CLI Command Reference.

## modify-db-proxy-endpoint

Il seguente esempio di codice mostra come utilizzare `modify-db-proxy-endpoint`.

## AWS CLI

Per modificare un endpoint proxy DB per un RDS database

L'`modify-db-proxy-endpoint` seguente modifica un endpoint proxy DB `proxyEndpoint` per impostare il timeout di lettura su 65 secondi.

```
aws rds modify-db-proxy-endpoint \  
  --db-proxy-endpoint-name proxyEndpoint \  
  --cli-read-timeout 65
```

Output:

```
{  
  "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"  
    }  
}
```

Per ulteriori informazioni, consulta [Modificare un endpoint proxy](#) nella Amazon RDS User Guide e [Modificare un endpoint proxy](#) nella Amazon Aurora User Guide.

- Per i API dettagli, consulta Command Reference. [ModifyDbProxyEndpoint](#) AWS CLI

## modify-db-proxy-target-group

Il seguente esempio di codice mostra come utilizzare `modify-db-proxy-target-group`.

### AWS CLI

Per modificare gli endpoint di un proxy DB

L'esempio seguente modifica un gruppo target di proxy DB per impostare il numero massimo di connessioni all'80% e il numero massimo di connessioni inattive al 10%.

```
aws rds modify-db-proxy-target-group \  
  --target-group-name default \  
  --db-proxy-name proxyExample \  
  --connection-pool-config MaxConnectionsPercent=80,MaxIdleConnectionsPercent=10
```

Output:

```
{  
  "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"  
    }  
}
```

Per ulteriori informazioni, consulta [Modifying an RDS Proxy](#) nella Amazon RDS User Guide e [Modifying an RDS Proxy](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta Command [ModifyDbProxyTargetGroup](#) Reference AWS CLI .

## modify-db-proxy

Il seguente esempio di codice mostra come utilizzare `modify-db-proxy`.

### AWS CLI

Per modificare un proxy DB per un RDS database

L'esempio seguente modifica un proxy DB denominato in `proxyExample` modo da richiedere SSL le relative connessioni.

```
aws rds modify-db-proxy \  
  --db-proxy-name proxyExample \  
  --require-tls
```

Output:

```
{  
  "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",
"UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
}
}
```

Per ulteriori informazioni, consulta [Modify an RDS Proxy](#) nella Amazon RDS User Guide e [Creazione di un RDS proxy](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta [ModifyDbProxy AWS CLI Command Reference](#).

## modify-db-shard-group

Il seguente esempio di codice mostra come utilizzare `modify-db-shard-group`.

### AWS CLI

Esempio 1: modificare un gruppo di shard DB

L'esempio seguente modifica la capacità massima di un gruppo di shard DB.

```
aws rds modify-db-shard-group \
  --db-shard-group-identifier my-db-shard-group \
  --max-acu 1000
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Amazon Aurora DB Clusters](#) nella Guida per l'utente di Amazon Aurora.

Esempio 2: per descrivere i tuoi gruppi di shard DB

L'execute-db-shard-groups comando seguente recupera i dettagli dei gruppi di shard DB dopo l'esecuzione del comando. modify-db-shard-group La capacità massima del gruppo di shard DB my-db-shard-group è ora di 1000 unità ACUs di capacità Aurora ().

```
aws rds describe-db-shard-groups
```

Output:

```

{
  "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-cekyceexample.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-cekyceexample.us-
east-2.rds.amazonaws.com"
    }
  ]
}

```

```

    }
  ]
}

```

Per ulteriori informazioni, consulta [Amazon Aurora DB Clusters](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [ModifyDbShardGroup](#)Reference AWS CLI .

## modify-db-snapshot-attribute

Il seguente esempio di codice mostra come utilizzare `modify-db-snapshot-attribute`.

### AWS CLI

Esempio 1: consentire a due AWS account di ripristinare un'istantanea del DB

L'`modify-db-snapshot-attribute` esempio seguente concede l'autorizzazione a due AWS account, con gli identificatori 111122223333 e 444455556666, per ripristinare lo snapshot DB denominato. `mydbsnapshot`

```

aws rds modify-db-snapshot-attribute \
  --db-snapshot-identifier mydbsnapshot \
  --attribute-name restore \
  --values-to-add {"111122223333","444455556666"}

```

Output:

```

{
  "DBSnapshotAttributesResult": {
    "DBSnapshotIdentifier": "mydbsnapshot",
    "DBSnapshotAttributes": [
      {
        "AttributeName": "restore",
        "AttributeValues": [
          "111122223333",
          "444455556666"
        ]
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [Sharing a Snapshot](#) nella Amazon RDS User Guide.

Esempio 2: per impedire a un AWS account di ripristinare uno snapshot DB

L'`modify-db-snapshot-attribute` seguente rimuove l'autorizzazione da un particolare AWS account per ripristinare lo snapshot DB denominato `mydbsnapshot`. Quando si specifica un singolo account, l'identificatore dell'account non può essere racchiuso tra virgolette o parentesi.

```
aws rds modify-db-snapshot-attribute \  
  --db-snapshot-identifier mydbsnapshot \  
  --attribute-name restore \  
  --values-to-remove 444455556666
```

Output:

```
{  
  "DBSnapshotAttributesResult": {  
    "DBSnapshotIdentifier": "mydbsnapshot",  
    "DBSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": [  
          "111122223333"  
        ]  
      }  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Sharing a Snapshot](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [ModifyDbSnapshotAttribute AWS CLI Command Reference](#).

## **modify-db-snapshot-attributes**

Il seguente esempio di codice mostra come utilizzare `modify-db-snapshot-attributes`.

### AWS CLI

Per modificare un attributo di snapshot del DB

L'`modify-db-snapshot-attribute` seguente consente di ripristinare due identificatori di AWS account 111122223333 e di ripristinare 444455556666 lo snapshot DB denominato. `mydbsnapshot`

```
aws rds modify-db-snapshot-attribute \  
  --db-snapshot-identifier mydbsnapshot \  
  --attribute-name restore \  
  --values-to-add '["111122223333", "444455556666"]'
```

Output:

```
{  
  "DBSnapshotAttributesResult": {  
    "DBSnapshotIdentifier": "mydbsnapshot",  
    "DBSnapshotAttributes": [  
      {  
        "AttributeName": "restore",  
        "AttributeValues": [  
          "111122223333",  
          "444455556666"  
        ]  
      }  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Sharing a Snapshot](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [ModifyDbSnapshotAttributes AWS CLI Command Reference](#).

## `modify-db-snapshot`

Il seguente esempio di codice mostra come utilizzare `modify-db-snapshot`.

### AWS CLI

Per modificare un'istantanea del DB

L'`modify-db-snapshot` seguente aggiorna un'istantanea di Postge SQL 10.6 denominata `Postge 11.7. db5-snapshot-upg-test` SQL La nuova versione del motore DB viene mostrata dopo che l'aggiornamento dello snapshot è terminato e il relativo stato è disponibile.

```
aws rds modify-db-snapshot \  
  --db-snapshot-identifier db5-snapshot-upg-test \  
  --engine-version 11.7
```

Output:

```
{  
  "DBSnapshot": {  
    "DBSnapshotIdentifier": "db5-snapshot-upg-test",  
    "DBInstanceIdentifier": "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Upgrade a PostgreSQL DB Snapshot](#) nella Amazon User Guide. RDS

- Per i API dettagli, consulta Command Reference. [ModifyDbSnapshot](#)AWS CLI

## modify-db-subnet-group

Il seguente esempio di codice mostra come utilizzare `modify-db-subnet-group`.

## AWS CLI

Per modificare un gruppo di sottoreti DB

L'`modify-db-subnet-group` seguente aggiunge una sottorete con l'ID `subnet-08e41f9e230222222` al gruppo di sottoreti DB denominato `mysubnetgroup`. Per mantenere le sottoreti esistenti nel gruppo di sottoreti, includete i relativi valori IDs as nell'opzione. `--subnet-ids` Assicuratevi di avere sottoreti con almeno due zone di disponibilità diverse nel gruppo di sottoreti DB.

```
aws rds modify-db-subnet-group \
  --db-subnet-group-name mysubnetgroup \
  --subnet-ids
  '["subnet-0a1dc4e1a6f123456", "subnet-070dd7ecb3aaaaaaa", "subnet-00f5b198bc0abcdef", "subnet-
```

Output:

```
{
  "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"
}
}

```

Per ulteriori informazioni, consulta la [Fase 3: Creare un gruppo di sottoreti DB](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta [ModifyDbSubnetGroup AWS CLI Command Reference](#).

## modify-event-subscription

Il seguente esempio di codice mostra come utilizzare `modify-event-subscription`.

### AWS CLI

Per modificare la sottoscrizione di un evento

L'esempio seguente disattiva l'abbonamento all'evento specificato, in modo che non pubblichi più notifiche sull'argomento Amazon Simple Notification Service specificato.

```

aws rds modify-event-subscription \
  --subscription-name my-instance-events \
  --no-enabled

```

Output:

```

{
  "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
  }
}

```

- Per API i dettagli, consulta AWS CLI Command [ModifyEventSubscriptionReference](#).

## modify-global-cluster

Il seguente esempio di codice mostra come utilizzare `modify-global-cluster`.

### AWS CLI

Per modificare un cluster DB globale

L'esempio seguente abilita la protezione da eliminazione per un cluster DB globale SQL compatibile con Aurora My.

```

aws rds modify-global-cluster \
  --global-cluster-identifier myglobalcluster \
  --deletion-protection

```

Output:

```

{
  "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": []
  }
}

```

Per ulteriori informazioni, consulta [Gestire un database globale Aurora](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta AWS CLI Command [ModifyGlobalCluster](#)Reference.

## promote-read-replica-db-cluster

Il seguente esempio di codice mostra come utilizzare `promote-read-replica-db-cluster`.

### AWS CLI

Per promuovere un cluster DB, leggi la replica

L'esempio seguente promuove la replica di lettura specificata in un cluster DB autonomo.

```

aws rds promote-read-replica-db-cluster \
  --db-cluster-identifier mydbcluster-1

```

Output:

```

{
  "DBCluster": {
    "AllocatedStorage": 1,
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1b",
      "us-east-1c"
    ],
    "BackupRetentionPeriod": 1,
    "DatabaseName": "",
    "DBClusterIdentifier": "mydbcluster-1",

```

```
    ...some output truncated...  
  }  
}
```

Per ulteriori informazioni, consulta [Promuovere una replica di lettura come cluster DB nella Guida per l'utente di Amazon Aurora](#).

- Per API i dettagli, consulta AWS CLI Command [PromoteReadReplicaDbCluster](#)Reference.

## promote-read-replica

Il seguente esempio di codice mostra come utilizzare `promote-read-replica`.

### AWS CLI

Per promuovere una replica di lettura

L'`promote-read-replica` esempio seguente promuove la replica di lettura specificata in modo che diventi un'istanza DB autonoma.

```
aws rds promote-read-replica \  
  --db-instance-identifier test-instance-repl
```

Output:

```
{  
  "DBInstance": {  
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance-repl",  
    "StorageType": "standard",  
    "ReadReplicaSourceDBInstanceIdentifier": "test-instance",  
    "DBInstanceStatus": "modifying",  
    ...some output truncated...  
  }  
}
```

- Per API i dettagli, vedere [PromoteReadReplica](#) in AWS CLI Command Reference.

## purchase-reserved-db-instance

Il seguente esempio di codice mostra come utilizzare `purchase-reserved-db-instance`.

## AWS CLI

Per acquistare un'offerta di istanze DB riservate

L'`purchase-reserved-db-instances-offering` seguente acquista un'offerta di istanze DB riservate. `reserved-db-instances-offering-id` Deve essere un ID di offerta valido, come restituito dal `describe-reserved-db-instances-offering` comando.

```
aws rds purchase-reserved-db-instances -offering -- reserved-db-instances-offering -id
438012d3-4a52-4cc7-b2e3-8dff72e0e706
```

- Per API i dettagli, vedere [PurchaseReservedDbInstance](#) in AWS CLI Command Reference.

## `purchase-reserved-db-instances-offerings`

Il seguente esempio di codice mostra come utilizzare `purchase-reserved-db-instances-offerings`.

### AWS CLI

Esempio 1: per trovare un'istanza DB riservata da acquistare

L'`describe-reserved-db-instances-offerings` seguente elenca le istanze MySQL DB riservate disponibili con la classe di istanze `db.t2.micro` e una durata di un anno. L'ID dell'offerta è necessario per acquistare un'istanza DB riservata.

```
aws rds describe-reserved-db-instances-offerings \
  --product-description mysql \
  --db-instance-class db.t2.micro \
  --duration 1
```

Output:

```
{
  "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 ...
]
}

```

Per ulteriori informazioni, consulta la sezione [Istanze database riservate per Amazon RDS nella Amazon RDS User Guide](#).

Esempio 2: per acquistare un'istanza DB riservata

L'purchase-reserved-db-instances-offeringesempio seguente mostra come acquistare l'offerta di istanze DB riservate dell'esempio precedente.

```
aws rds purchase-reserved-db-instances -offering -- reserved-db-instances-offering -id 8ba30be1-b9ec-447f-8f23-6114e3f4c7b4
```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta la sezione [Istanze database riservate per Amazon RDS nella Amazon RDS User Guide](#).

- Per API i dettagli, consulta [PurchaseReservedDbInstancesOfferings AWS CLI Command Reference](#).

## reboot-db-instance

Il seguente esempio di codice mostra come utilizzare `reboot-db-instance`.

### AWS CLI

Per riavviare un'istanza DB

L'`reboot-db-instance` esempio seguente avvia il riavvio dell'istanza DB specificata.

```

aws rds reboot-db-instance \
  --db-instance-identifier test-mysql-instance

```

Output:

```

{
  "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...  
}  
}
```

Per ulteriori informazioni, consulta [Rebooting a DB Instance](#) nella Amazon RDS User Guide.

- Per API i dettagli, vedere [RebootDBInstance](#) in AWS CLI Command Reference.

## reboot-db-shard-group

Il seguente esempio di codice mostra come usare `reboot-db-shard-group`.

### AWS CLI

Esempio 1: riavviare un gruppo di shard DB

L'`reboot-db-shard-group`esempio seguente riavvia un gruppo di shard DB.

```
aws rds reboot-db-shard-group \  
--db-shard-group-identifier my-db-shard-group
```

Output:

```
{  
  "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-cekyexample.us-  
east-2.rds.amazonaws.com"  
    }  
  ]  
}
```

```
}
```

Per ulteriori informazioni, consulta [Riavvio di un cluster Amazon Aurora DB o di un'istanza Amazon Aurora DB nella Amazon Aurora User Guide](#).

Esempio 2: per descrivere i tuoi gruppi di shard DB

L'output del comando `aws rds describe-db-shard-groups` seguente recupera i dettagli dei gruppi di shard DB dopo l'esecuzione del comando `aws rds reboot-db-shard-group` Il gruppo di shard DB `my-db-shard-group` è ora in fase di riavvio.

```
aws rds describe-db-shard-groups
```

Output:

```
{
  "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-cekyceexample.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-cekyceexample.us-east-2.rds.amazonaws.com"
    }
  ]
}
```



Per ulteriori informazioni, consulta [Riavvio di un cluster Amazon Aurora DB o di un'istanza Amazon Aurora DB nella Amazon Aurora User Guide](#).

- Per i API dettagli, consulta Command Reference. [RebootDbShardGroup](#) AWS CLI

## register-db-proxy-targets

Il seguente esempio di codice mostra come utilizzare `register-db-proxy-targets`.

### AWS CLI

Per registrare un proxy DB con un database

L'`register-db-proxy-targets` esempio seguente crea l'associazione tra un database e un proxy.

```
aws rds register-db-proxy-targets \  
  --db-proxy-name proxyExample \  
  --db-cluster-identifiers database-5
```

Output:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

```
]
}
```

Per ulteriori informazioni, consulta [Creazione di un RDS proxy](#) nella Amazon RDS User Guide e [Creazione di un RDS proxy](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta [RegisterDbProxyTargets AWS CLI Command Reference](#).

## **remove-from-global-cluster**

Il seguente esempio di codice mostra come utilizzare `remove-from-global-cluster`.

### AWS CLI

Per scollegare un cluster secondario Aurora da un cluster di database globale Aurora

L'esempio seguente scollega un cluster secondario Aurora da un cluster di database globale Aurora. Il cluster passa da un cluster di sola lettura a un cluster autonomo con funzionalità di lettura/scrittura.

```
aws rds remove-from-global-cluster \
  --region us-west-2 \
  --global-cluster-identifier myglobalcluster \
  --db-cluster-identifier arn:aws:rds:us-west-2:123456789012:cluster:DB-1
```

Output:

```
{
  "GlobalCluster": {
    "GlobalClusterIdentifier": "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"
    }
]
}
}

```

Per ulteriori informazioni, consulta [Rimuovere un cluster da un database globale di Amazon Aurora](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta AWS CLI Command [RemoveFromGlobalCluster](#)Reference.

## remove-option-from-option-group

Il seguente esempio di codice mostra come utilizzare `remove-option-from-option-group`.

### AWS CLI

Per eliminare un'opzione da un gruppo di opzioni

L'`remove-option-from-option-group`esempio seguente rimuove l'`OEM`opzione da `myoptiongroup`.

```

aws rds remove-option-from-option-group \
  --option-group-name myoptiongroup \
  --options OEM \
  --apply-immediately

```

Output:

```

{
  "OptionGroup": {
    "OptionGroupName": "myoptiongroup",
    "OptionGroupDescription": "Test",

```

```
    "EngineName": "oracle-ee",
    "MajorEngineVersion": "19",
    "Options": [],
    "AllowsVpcAndNonVpcInstanceMemberships": true,
    "OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:myoptiongroup"
  }
}
```

Per ulteriori informazioni, consulta [Rimuovere un'opzione da un gruppo di opzioni](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta [RemoveOptionFromOptionGroup AWS CLI Command Reference](#).

## **remove-role-from-db-cluster**

Il seguente esempio di codice mostra come utilizzare `remove-role-from-db-cluster`.

### AWS CLI

Per dissociare un ruolo AWS Identity and Access Management (IAM) da un cluster DB

L'`remove-role-from-db-cluster` seguente rimuove un ruolo da un cluster DB.

```
aws rds remove-role-from-db-cluster \
  --db-cluster-identifier mydbcluster \
  --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Associare un IAM ruolo a un cluster Amazon Aurora SQL My DB](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta Command [RemoveRoleFromDbCluster](#) Reference AWS CLI .

## **remove-role-from-db-instance**

Il seguente esempio di codice mostra come utilizzare `remove-role-from-db-instance`.

### AWS CLI

Per dissociare un ruolo AWS Identity and Access Management (IAM) da un'istanza DB

L'`remove-role-from-db-instance` seguente rimuove il ruolo denominato `rds-s3-integration-role` da un'istanza Oracle DB denominata `test-instance`.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Disabling RDS SQL Server Integration with S3](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [RemoveRoleFromDbInstance](#) Reference.

## `remove-source-identifier-from-subscription`

Il seguente esempio di codice mostra come utilizzare `remove-source-identifier-from-subscription`.

### AWS CLI

Per rimuovere un identificatore di origine da un abbonamento

L'`remove-source-identifier` seguente rimuove l'identificatore di origine specificato da un abbonamento esistente.

```
aws rds remove-source-identifier-from-subscription \  
  --subscription-name my-instance-events \  
  --source-identifier test-instance-repl
```

Output:

```
{  
  "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
}
}

```

- Per API i dettagli, vedere [RemoveSourceIdentifierFromSubscription](#) in AWS CLI Command Reference.

## remove-tags-from-resource

Il seguente esempio di codice mostra come utilizzare `remove-tags-from-resource`.

### AWS CLI

Per rimuovere i tag da una risorsa

L'esempio seguente rimuove i tag da una risorsa.

```

aws rds remove-tags-from-resource \
  --resource-name arn:aws:rds:us-east-1:123456789012:db:mydbinstance \
  --tag-keys Name Environment

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging Amazon RDS resources](#) nella Amazon RDS User Guide e [Tagging Amazon RDS resources nella Amazon Aurora User Guide](#).

- Per API i dettagli, consulta Command [RemoveTagsFromResource](#) Reference AWS CLI .

## reset-db-cluster-parameter-group

Il seguente esempio di codice mostra come utilizzare `reset-db-cluster-parameter-group`.

## AWS CLI

Esempio 1: per ripristinare tutti i parametri ai valori predefiniti

L'`reset-db-cluster-parameter-group` seguente reimposta tutti i valori dei parametri in un gruppo di parametri del cluster DB creato dal cliente ai valori predefiniti.

```
aws rds reset-db-cluster-parameter-group \  
  --db-cluster-parameter-group-name mydbclpg \  
  --reset-all-parameters
```

Output:

```
{  
  "DBClusterParameterGroupName": "mydbclpg"  
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di parametri DB e i gruppi di parametri del cluster DB](#) nella Guida per l'utente di Amazon Aurora.

Esempio 2: per ripristinare i valori predefiniti di parametri specifici

L'`reset-db-cluster-parameter-group` seguente reimposta i valori dei parametri per parametri specifici ai valori predefiniti in un gruppo di parametri del cluster DB creato dal cliente.

```
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"
```

Output:

```
{  
  "DBClusterParameterGroupName": "mydbclpgy"  
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di parametri DB e i gruppi di parametri del cluster DB](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta [ResetDbClusterParameterGroup AWS CLI Command Reference](#).

## reset-db-parameter-group

Il seguente esempio di codice mostra come utilizzare `reset-db-parameter-group`.

### AWS CLI

Esempio 1: per ripristinare tutti i parametri ai valori predefiniti

L'`reset-db-parameter-group` seguente reimposta tutti i valori dei parametri in un gruppo di parametri DB creato dal cliente ai valori predefiniti.

```
aws rds reset-db-parameter-group \  
  --db-parameter-group-name mypg \  
  --reset-all-parameters
```

Output:

```
{  
  "DBParameterGroupName": "mypg"  
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di parametri DB](#) nella Amazon RDS User Guide e [Lavorare con i gruppi di parametri DB e i gruppi di parametri del cluster DB](#) nella Guida per l'utente di Amazon Aurora.

Esempio 2: per ripristinare i valori predefiniti di parametri specifici

L'`reset-db-parameter-group` seguente reimposta i valori dei parametri per parametri specifici ai valori predefiniti in un gruppo di parametri DB creato dal cliente.

```
aws rds reset-db-parameter-group \  
  --db-parameter-group-name mypg \  
  --parameters "ParameterName=max_connections,ApplyMethod=immediate" \  
               "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

Output:

```
{
```



```
"DBParameterGroupName": "mypg"
}
```

Per ulteriori informazioni, consulta [Lavorare con i gruppi di parametri DB](#) nella Amazon RDS User Guide e [Lavorare con i gruppi di parametri DB e i gruppi di parametri del cluster DB](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta [ResetDbParameterGroup AWS CLI Command Reference](#).

## restore-db-cluster-from-s3

Il seguente esempio di codice mostra come utilizzare `restore-db-cluster-from-s3`.

### AWS CLI

Per ripristinare un cluster Amazon Aurora DB da Amazon S3

L'esempio seguente ripristina un cluster DB compatibile con Amazon Aurora SQL My versione 5.7 da un file di backup SQL My 5.7 DB in Amazon S3.

```
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
```

Output:

```
{
  "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": "Z1PVIF0EXAMPLE",
    "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": []
  }
}

```

Per ulteriori informazioni, consulta [Migrazione dei dati SQL da My utilizzando un bucket Amazon S3](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta [RestoreDbClusterFromS3](#) in Command Reference.AWS CLI

## restore-db-cluster-from-snapshot

Il seguente esempio di codice mostra come usare `restore-db-cluster-from-snapshot`.

### AWS CLI

Per ripristinare un cluster DB da un'istantanea

Quanto segue `restore-db-cluster-from-snapshot` ripristina un cluster Aurora SQL Postgre DB compatibile con SQL Postgre versione 10.7 da uno snapshot del cluster DB denominato `test-instance-snapshot`

```
aws rds restore-db-cluster-from-snapshot \  
  --db-cluster-identifier newdbcluster \  
  --snapshot-identifier test-instance-snapshot \  
  --engine aurora-postgresql \  
  --engine-version 10.7
```

Output:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-west-2c",  
      "us-west-2a",  
      "us-west-2b"  
    ],  
    "BackupRetentionPeriod": 7,  
    "DatabaseName": "",  
    "DBClusterIdentifier": "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": []
  }
}

```

Per ulteriori informazioni, consulta [Restoring from a DB Cluster Snapshot](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta Command [RestoreDbClusterFromSnapshot](#) Reference AWS CLI .

## restore-db-cluster-to-point-in-time

Il seguente esempio di codice mostra come utilizzare `restore-db-cluster-to-point-in-time`.

### AWS CLI

Per ripristinare un cluster DB a un'ora specificata

L'esempio seguente ripristina il cluster DB denominato `database-4` all'ora più recente possibile. L'utilizzo `copy-on-write` come tipo di ripristino ripristina il nuovo cluster DB come clone del cluster DB di origine.

```
aws rds restore-db-cluster-to-point-in-time \  
  --source-db-cluster-identifier database-4 \  
  --db-cluster-identifier sample-cluster-clone \  
  --restore-type copy-on-write \  
  --use-latest-restorable-time
```

Output:

```
{  
  "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": "Z1PVIF0EXAMPLE",  
    "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
  }
}

```

Per ulteriori informazioni, consulta [Ripristino di un cluster DB a un'ora specificata nella Guida](#) per l'utente di Amazon Aurora.

- Per API i dettagli, consulta AWS CLI Command [RestoreDbClusterToPointInTime](#)Reference.

## restore-db-instance-from-db-snapshot

Il seguente esempio di codice mostra come utilizzare `restore-db-instance-from-db-snapshot`.

### AWS CLI

Per ripristinare un'istanza DB da un'istantanea del database

L'esempio seguente crea una nuova istanza DB denominata `db7-new-instance` con la classe di istanza `db.t3.small` DB dallo snapshot DB specificato. L'istanza DB di origine da cui è stata scattata l'istantanea utilizza una classe di istanza DB obsoleta, quindi non è possibile aggiornarla.

```

aws rds restore-db-instance-from-db-snapshot \
  --db-instance-identifier db7-new-instance \
  --db-snapshot-identifier db7-test-snapshot \
  --db-instance-class db.t3.small

```

Output:

```
{
  "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": []
  }
}
```

Per ulteriori informazioni, consulta [Restoring from a DB Snapshot](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [RestoreDbInstanceFromDbSnapshot](#) Reference.

### **restore-db-instance-from-s3**

Il seguente esempio di codice mostra come utilizzare `restore-db-instance-from-s3`.

#### AWS CLI

Per ripristinare un'istanza DB da un backup in Amazon S3

L'esempio seguente crea una nuova istanza DB denominata `restored-test-instance` da un backup esistente nel bucket `my-backups` S3.

```
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
```

- Per API i dettagli, consulta [RestoreDbInstanceFromS3](#) in AWS CLI Command Reference.

## restore-db-instance-to-point-in-time

Il seguente esempio di codice mostra come usare `restore-db-instance-to-point-in-time`.

### AWS CLI

Esempio 1: ripristinare un'istanza DB in un determinato momento

L'esempio seguente ripristina `test-instance` una nuova istanza DB denominata `restored-test-instance`, a partire dall'ora specificata.

```
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
```

Output:

```
{  
  "DBInstance": {  
    "AllocatedStorage": 20,  
    "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:restored-test-  
instance",  
    "DBInstanceStatus": "creating",  
    "DBInstanceIdentifier": "restored-test-instance",  
    ...some output omitted...  
  }  
}
```

Per ulteriori informazioni, consulta [Ripristino di un'istanza DB a un'ora specificata](#) nella Amazon RDS User Guide.



## Esempio 2: ripristinare un'istanza DB a un'ora specificata da un backup replicato

L'restore-db-instance-to-point-in-time esempio seguente ripristina un'istanza DB Oracle all'ora specificata da un backup automatico replicato.

```
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
```

Output:

```
{  
  "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-KGLXG75BGVIWKQT7NQ4EXAMPLE",  
    "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": []  
  }  
}
```

Per ulteriori informazioni, consulta [Ripristino a un'ora specificata da un backup replicato](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [RestoreDbInstanceToPointInTimeReference](#).

## start-activity-stream

Il seguente esempio di codice mostra come utilizzare `start-activity-stream`.

### AWS CLI

Per avviare un flusso di attività del database

L'esempio seguente avvia un flusso di attività asincrono per monitorare un cluster Aurora denominato. `my-pg-cluster`

```
aws rds start-activity-stream \  
  --region us-east-1 \  
  --mode async \  
  --kms-key-id arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-h123-456i789jk0l1 \  
  --resource-arn arn:aws:rds:us-east-1:1234567890123:cluster:my-pg-cluster \  
  --apply-immediately
```

Output:

```
{  
  "KmsKeyId": "arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-h123-456i789jk0l1",  
  "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGH11JKLM2NOPQ3R4S",  
  "Status": "starting",  
  "Mode": "async",  
  "ApplyImmediately": true  
}
```

Per ulteriori informazioni, consulta [Avvio di un flusso di attività del database](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta [StartActivityStream AWS CLI Command Reference](#).

## start-db-cluster

Il seguente esempio di codice mostra come utilizzare `start-db-cluster`.

## AWS CLI

Per avviare un cluster DB

L'`start-db-cluster` seguente avvia un cluster DB e le relative istanze DB.

```
aws rds start-db-cluster \  
  --db-cluster-identifier mydbcluster
```

Output:

```
{  
  "DBCluster": {  
    "AllocatedStorage": 1,  
    "AvailabilityZones": [  
      "us-east-1a",  
      "us-east-1e",  
      "us-east-1b"  
    ],  
    "BackupRetentionPeriod": 1,  
    "DatabaseName": "mydb",  
    "DBClusterIdentifier": "mydbcluster",  
    "...some output truncated..."  
  }  
}
```

Per ulteriori informazioni, consulta [Arresto e avvio di un cluster Amazon Aurora DB](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [StartDbCluster](#) Reference AWS CLI .

## **start-db-instance-automated-backups-replication**

Il seguente esempio di codice mostra come utilizzare `start-db-instance-automated-backups-replication`.

## AWS CLI

Per abilitare i backup automatici tra regioni

L'`start-db-instance-automated-backups-replication` seguente replica i backup automatici da un'istanza DB nella regione Stati Uniti orientali (Virginia settentrionale) a Stati Uniti occidentali (Oregon). Il periodo di conservazione dei backup è di 14 giorni.

```
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
```

Output:

```
{  
  "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-jkib2gfgq5rv7replzadtausbrktni2bn4example"  
  }  
}
```

Per ulteriori informazioni, consulta la [sezione Abilitazione dei backup automatici tra regioni](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [StartDbInstanceAutomatedBackupsReplication](#) Reference.

## start-db-instance

Il seguente esempio di codice mostra come utilizzare `start-db-instance`.

### AWS CLI

Per avviare un'istanza DB

L'`start-db-instance`esempio seguente avvia l'istanza DB specificata.

```
aws rds start-db-instance \  
  --db-instance-identifier test-instance
```

Output:

```
{  
  "DBInstance": {  
    "DBInstanceStatus": "starting",  
    ...some output truncated...  
  }  
}
```

- Per API i dettagli, vedere [StartDbInstance](#) in AWS CLI Command Reference.

## start-export-task

Il seguente esempio di codice mostra come utilizzare `start-export-task`.

### AWS CLI

Per esportare uno snapshot in Amazon S3

L'`start-export-task`esempio seguente esporta uno snapshot DB denominato nel `db5-snapshot-test` bucket Amazon S3 denominato `mybucket`

```
aws rds start-export-task \  
  --export-task-identifier 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
```

**Output:**

```
{
  "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",
  "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
}
```

Per ulteriori informazioni, consulta [Esportazione di uno snapshot in un bucket Amazon S3](#) nella Amazon User Guide. RDS

- Per i API dettagli, consulta [StartExportTask](#) Command Reference. AWS CLI

**stop-activity-stream**

Il seguente esempio di codice mostra come utilizzare `stop-activity-stream`.

**AWS CLI**

Per interrompere un flusso di attività del database

L'esempio seguente interrompe un flusso di attività in un cluster Aurora denominato `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
```

**Output:**

```
{
  "KmsKeyId": "arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-
h123-456i789jk0l1",
  "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGH11JKLM2NOPQ3R4S",
}
```

```
"Status": "stopping"
}
```

Per ulteriori informazioni, consulta [Interruzione di un flusso di attività nella Guida](#) per l'utente di Amazon Aurora.

- Per API i dettagli, consulta AWS CLI Command [StopActivityStream](#)Reference.

## stop-db-cluster

Il seguente esempio di codice mostra come utilizzare `stop-db-cluster`.

### AWS CLI

Per arrestare un cluster DB

L'`stop-db-cluster` esempio seguente arresta un cluster DB e le relative istanze DB.

```
aws rds stop-db-cluster \
  --db-cluster-identifier mydbcluster
```

Output:

```
{
  "DBCluster": {
    "AllocatedStorage": 1,
    "AvailabilityZones": [
      "us-east-1a",
      "us-east-1e",
      "us-east-1b"
    ],
    "BackupRetentionPeriod": 1,
    "DatabaseName": "mydb",
    "DBClusterIdentifier": "mydbcluster",
    ...some output truncated...
  }
}
```

Per ulteriori informazioni, consulta [Arresto e avvio di un cluster Amazon Aurora DB](#) nella Guida per l'utente di Amazon Aurora.

- Per API i dettagli, consulta Command [StopDbCluster](#)Reference AWS CLI .

## stop-db-instance-automated-backups-replication

Il seguente esempio di codice mostra come utilizzare `stop-db-instance-automated-backups-replication`.

### AWS CLI

Per interrompere la replica dei backup automatici

Quando segue `stop-db-instance-automated-backups-replication` termina la replica dei backup automatici nella regione Stati Uniti occidentali (Oregon). I backup replicati vengono conservati in base al periodo di conservazione dei backup impostato.

```
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"
```

Output:

```
{  
  "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-jkib2gfg5rv7replzadausbrktni2bn4example"
    }
}
```

Per ulteriori informazioni, consulta [Stopping automation backup replication](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [StopDbInstanceAutomatedBackupsReplication](#) Reference.

## stop-db-instance

Il seguente esempio di codice mostra come utilizzare `stop-db-instance`.

### AWS CLI

Per interrompere un'istanza DB

L'esempio seguente arresta l'istanza DB specificata.

```
aws rds stop-db-instance \
  --db-instance-identifier test-instance
```

Output:

```
{
  "DBInstance": {
    "DBInstanceStatus": "stopping",
    ...some output truncated...
  }
}
```

- Per API i dettagli, vedere [StopDbInstance](#) in AWS CLI Command Reference.

## switchover-blue-green-deployment

Il seguente esempio di codice mostra come utilizzare `switchover-blue-green-deployment`.

### AWS CLI

Esempio 1: cambiare una distribuzione blu/verde per un'istanza RDS

L'operazione `switchover-blue-green-deployment` seguente promuove l'ambiente verde specificato come nuovo ambiente di produzione.

```
aws rds switchover-blue-green-deployment \  
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake \  
  --switchover-timeout 300
```

Output:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

```

        "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"
  }
}

```

Per ulteriori informazioni, consulta [Switching a blue/green deployment](#) nella Amazon RDS User Guide.

Esempio 2: promuovere una distribuzione blu/verde per un cluster Aurora My DB SQL

L'`switchover-blue-green-deployment` seguente promuove l'ambiente verde specificato come nuovo ambiente di produzione.

```

aws rds switchover-blue-green-deployment \
  --blue-green-deployment-identifier bgd-wi89nwzglccsfake \
  --switchover-timeout 300

```

Output:

```

{
  "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-nplikl",
        "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-miszl",
        "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"
}
}

```

Per ulteriori informazioni, consulta [Switching a blu/green deployment](#) nella Amazon Aurora User Guide.

- Per API i dettagli, consulta Command [SwitchoverBlueGreenDeployment](#)Reference AWS CLI .

## Esempi di utilizzo di Amazon RDS Data Service AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface con Amazon RDS Data Service.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **batch-execute-statement**

Il seguente esempio di codice mostra come utilizzare `batch-execute-statement`.

#### AWS CLI

Per eseguire un'istruzione batch

L'esempio seguente esegue un'istruzione batch su una matrice di dati con un set di parametri.

```
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"}}]]"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Using the Data API for Aurora Serverless](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [BatchExecuteStatement](#) Reference.

### **begin-transaction**

Il seguente esempio di codice mostra come utilizzare `begin-transaction`.

## AWS CLI

Per iniziare una SQL transazione

L'`begin-transaction` esempio seguente avvia una SQL transazione.

```
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"
```

Output:

```
{  
  "transactionId": "ABC1234567890xyz"  
}
```

Per ulteriori informazioni, consulta [Using the Data API for Aurora Serverless](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [BeginTransaction](#) Reference.

## `commit-transaction`

Il seguente esempio di codice mostra come utilizzare `commit-transaction`.

## AWS CLI

Per effettuare una SQL transazione

L'`commit-transaction` esempio seguente termina la SQL transazione specificata e conferma le modifiche apportate nell'ambito di essa.

```
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"
```

Output:

```
{  
  "transactionStatus": "Transaction Committed"  
}
```

```
}

```

Per ulteriori informazioni, consulta [Using the Data API for Aurora Serverless](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [CommitTransaction](#) Reference.

## execute-statement

Il seguente esempio di codice mostra come utilizzare `execute-statement`.

### AWS CLI

Esempio 1: eseguire un'istruzione SQL che fa parte di una transazione

L'esempio seguente esegue un'istruzione SQL che fa parte di una transazione.

```
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"
```

Output:

```
{
  "numberOfRecordsUpdated": 1
}
```

Esempio 2: eseguire un'istruzione SQL con parametri

L'esempio seguente esegue un'istruzione SQL con parametri.

```
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\"}}]"
```



Output:

```
{
  "numberOfRecordsUpdated": 1
}
```

Per ulteriori informazioni, consulta [Using the Data API for Aurora Serverless](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [ExecuteStatement](#)Reference.

## rollback-transaction

Il seguente esempio di codice mostra come utilizzare `rollback-transaction`.

AWS CLI

Per ripristinare una SQL transazione

L'`rollback-transaction`esempio seguente ripristina la SQL transazione specificata.

```
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"
```

Output:

```
{
  "transactionStatus": "Rollback Complete"
}
```

Per ulteriori informazioni, consulta [Using the Data API for Aurora Serverless](#) nella Amazon RDS User Guide.

- Per API i dettagli, consulta AWS CLI Command [RollbackTransaction](#)Reference.

## Esempi di utilizzo di Amazon RDS Performance Insights AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon RDS Performance Insights. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **describe-dimension-keys**

Il seguente esempio di codice mostra come utilizzare `describe-dimension-keys`.

AWS CLI

Per descrivere le chiavi dimensionali

Questo esempio richiede i nomi di tutti gli eventi di attesa. I dati vengono riepilogati in base al nome dell'evento e ai valori aggregati di tali eventi nel periodo di tempo specificato.

Comando:

```
aws pi describe-dimension-keys --service-type RDS --identifier db-LKCG0BK26374TPTDFX0IWVCPM --start-time 1527026400 --end-time 1527080400 --metric db.load.avg --group-by '{"Group": "db.wait_event"}'
```

Output:

```
{
  "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
    }
  ]
}

```

- Per API i dettagli, vedere [DescribeDimensionKeys](#) in AWS CLI Command Reference.

## get-resource-metrics

Il seguente esempio di codice mostra come utilizzare `get-resource-metrics`.

### AWS CLI

Per ottenere le metriche delle risorse

Questo esempio richiede punti dati per il gruppo di dimensioni `db.wait_event` e per la dimensione `db.wait_event.name` all'interno di quel gruppo. Nella risposta, i punti dati pertinenti vengono raggruppati in base alla dimensione richiesta (`db.wait_event.name`).

Comando:

```
aws pi get-resource-metrics --service-type RDS --identifier db-LKCG0BK26374TPTDFX0IWVCP  
PM --start-time 1527026400 --end-time 1527080400 --period-  
in-seconds 300 --metric db.load.avg --metric-queries file://metric-queries.json
```

Gli argomenti per sono memorizzati in un file, `--metric-queries` JSON `metric-queries.json` Ecco i contenuti del file:

```
[  
  {  
    "Metric": "db.load.avg",  
    "GroupBy": {  
      "Group": "db.wait_event"  
    }  
  }  
]
```

Output:

```
{  
  "AlignedEndTime": 1.5270804E9,  
  "AlignedStartTime": 1.5270264E9,  
  "Identifier": "db-LKCG0BK26374TPTDFX0IWVCP",  
  "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...>
]
}

```

- Per API i dettagli, vedere [GetResourceMetrics](#) in AWS CLI Command Reference.

## Esempi di utilizzo di Amazon Redshift AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Redshift. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **accept-reserved-node-exchange**

Il seguente esempio di codice mostra come utilizzare `accept-reserved-node-exchange`.

AWS CLI

Accettare lo scambio di nodi riservati

L'esempio seguente accetta lo scambio di un nodo DC1 riservato con un nodo DC2 riservato.

```
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
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Upgrade Reserved Nodes With the AWS CLI](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [AcceptReservedNodeExchange](#) Reference AWS CLI .

## authorize-cluster-security-group-ingress

Il seguente esempio di codice mostra come utilizzare `authorize-cluster-security-group-ingress`.

### AWS CLI

L' GroupThis esempio di autorizzazione dell'accesso a una EC2 sicurezza autorizza l'accesso a un gruppo di EC2 sicurezza Amazon denominato. Comando:

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-owner-id 123445677890
```

L'autorizzazione dell'accesso a un CIDR rangeThis esempio autorizza l'accesso a un intervallo. Comando: CIDR

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name mysecuritygroup --cidrip 192.168.100.100/32
```

- Per i API dettagli, vedere [AuthorizeClusterSecurityGroupIngress](#) in Command Reference.AWS CLI

## authorize-snapshot-access

Il seguente esempio di codice mostra come utilizzare `authorize-snapshot-access`.

## AWS CLI

Autorizza un AWS account al ripristino Un Snapshot. Questo esempio autorizza l'AWS account 444455556666 a ripristinare l'istantanea. `my-snapshot-id`. Per impostazione predefinita, l'output è in formato JSON.

```
aws redshift authorize-snapshot-access --snapshot-id my-snapshot-id --account-with-restore-access 444455556666
```

### Risultato:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [AuthorizeSnapshotAccess](#) in AWS CLI Command Reference.

## batch-delete-cluster-snapshots

Il seguente esempio di codice mostra come utilizzare `batch-delete-cluster-snapshots`.



## AWS CLI

Per eliminare un set di istantanee del cluster

L'`batch-delete-cluster-snapshots` seguente elimina un set di istantanee manuali del cluster.

```
aws redshift batch-delete-cluster-snapshots \  
  --  
  identifiers SnapshotIdentifier=mycluster-2019-11-06-14-12 SnapshotIdentifier=mycluster-2019-
```

Output:

```
{  
  "Resources": [  
    "mycluster-2019-11-06-14-12",  
    "mycluster-2019-11-06-14-20"  
  ]  
}
```

Per ulteriori informazioni, consulta [Amazon Redshift Snapshots](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [BatchDeleteClusterSnapshots](#) Reference AWS CLI .

## `batch-modify-cluster-snapshots`

Il seguente esempio di codice mostra come utilizzare `batch-modify-cluster-snapshots`.

## AWS CLI

Per modificare un set di istantanee del cluster

L'`batch-modify-cluster-snapshots` seguente modifica le impostazioni per un set di istantanee del cluster.

```
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
```

**Output:**

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Amazon Redshift Snapshots](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [BatchModifyClusterSnapshots](#)Reference AWS CLI .

**cancel-resize**

Il seguente esempio di codice mostra come utilizzare `cancel-resize`.

**AWS CLI**

Per annullare il ridimensionamento di un cluster

L'esempio seguente annulla una classica operazione di ridimensionamento per un cluster.

```
aws redshift cancel-resize \
  --cluster-identifier mycluster
```

**Output:**

```
{
  "TargetNodeType": "dc2.large",
  "TargetNumberOfNodes": 2,
  "TargetClusterType": "multi-node",
  "Status": "CANCELLING",
  "ResizeType": "ClassicResize",
  "TargetEncryptionType": "NONE"
}
```

Per ulteriori informazioni, consulta [Resizing Clusters in Amazon Redshift nella Amazon Redshift Cluster Management Guide](#).

- Per i API dettagli, consulta Command Reference. [CancelResize](#) AWS CLI

## copy-cluster-snapshot

Il seguente esempio di codice mostra come utilizzare `copy-cluster-snapshot`.

### AWS CLI

L' `aws redshift copy-cluster-snapshot` restituisce una descrizione di tutte le versioni del cluster. Per impostazione predefinita, l'output è in JSON formato. Comando:

```
aws redshift copy-cluster-snapshot --source-snapshot-identifier
cm:examplecluster-2013-01-22-19-27-58 --target-snapshot-identifier my-saved-
snapshot-copy
```

### Risultato:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [CopyClusterSnapshot](#) in AWS CLI Command Reference.

## create-cluster-parameter-group

Il seguente esempio di codice mostra come utilizzare `create-cluster-parameter-group`.

### AWS CLI

L' GroupThis esempio di creazione di un parametro cluster crea un nuovo gruppo di parametri del cluster. Comando:

```
aws redshift create-cluster-parameter-group --parameter-group-name
myclusterparametergroup --parameter-group-family redshift-1.0 --description "My
first cluster parameter group"
```

### Risultato:

```
{
  "ClusterParameterGroup": {
    "ParameterGroupFamily": "redshift-1.0",
    "Description": "My first cluster parameter group",
    "ParameterGroupName": "myclusterparametergroup"
  },
  "ResponseMetadata": {
    "RequestId": "739448f0-64cc-11e2-8f7d-3b939af52818"
  }
}
```

- Per API i dettagli, vedere [CreateClusterParameterGroup](#) in AWS CLI Command Reference.

## create-cluster-security-group

Il seguente esempio di codice mostra come utilizzare `create-cluster-security-group`.

## AWS CLI

La creazione di un GroupThis esempio di Cluster Security crea un nuovo gruppo di sicurezza del cluster. Per impostazione predefinita, l'output è in JSON formato. Comando:

```
aws redshift create-cluster-security-group --cluster-security-group-name
mysecuritygroup --description "This is my cluster security group"
```

Risultato:

```
{
  "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"
    }
  }
}
```

È inoltre possibile ottenere le stesse informazioni in formato testo utilizzando l'opzione. Comando:

```
--output text
```

Opzione.comando:

```
aws redshift create-cluster-security-group --cluster-security-group-name
mysecuritygroup --description "This is my cluster security group" --output text
```

Risultato:

```
This is my cluster security group 300454760768 mysecuritygroup
```

```
a0c0bfab-343a-11e2-95d2-c3dc9fe8ab57
```

- Per API i dettagli, vedere [CreateClusterSecurityGroup](#) in AWS CLI Command Reference.

## create-cluster-snapshot

Il seguente esempio di codice mostra come utilizzare `create-cluster-snapshot`.

### AWS CLI

L' `Snapshot` Questo esempio di creazione di un cluster crea una nuova istantanea del cluster. Per impostazione predefinita, l'output è in JSON formato. Comando:

```
aws redshift create-cluster-snapshot --cluster-identifier mycluster --snapshot-identifier my-snapshot-id
```

### Risultato:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [CreateClusterSnapshot](#) in AWS CLI Command Reference.

## create-cluster-subnet-group

Il seguente esempio di codice mostra come utilizzare `create-cluster-subnet-group`.

### AWS CLI

L' GroupThis esempio di creazione di una sottorete del cluster crea un nuovo gruppo di sottoreti del cluster. Comando:

```
aws redshift create-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
--description "My subnet group" --subnet-ids subnet-763fdd1c
```

### Risultato:

```
{
  "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"
  }
}
```

- Per i API dettagli, vedere [CreateClusterSubnetGroup](#) in Command Reference.AWS CLI

## create-cluster

Il seguente esempio di codice mostra come utilizzare `create-cluster`.

## AWS CLI

L' esempio `aws redshift create-cluster --node-type dw.hs1.xlarge --number-of-nodes 2 --master-username adminuser --master-user-password TopSecret1 --cluster-identifier mycluster` crea un cluster con il set minimo di parametri. Per impostazione predefinita, l'output è in JSON formato.

```
aws redshift create-cluster --node-type dw.hs1.xlarge --number-of-nodes 2 --master-username adminuser --master-user-password TopSecret1 --cluster-identifier mycluster
```

Risultato:

```
{
  "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"
  }
}
```



- Per API i dettagli, vedere [CreateCluster](#) in AWS CLI Command Reference.

## create-event-subscription

Il seguente esempio di codice mostra come utilizzare `create-event-subscription`.

### AWS CLI

Per creare una sottoscrizione alle notifiche per un evento

L'`create-event-subscription` esempio seguente crea una sottoscrizione per la notifica degli eventi.

```
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
```

Output:

```
{  
  "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": []  
  }  
}
```

Per ulteriori informazioni, consulta la sezione [Abbonamento alle notifiche degli eventi di Amazon Redshift](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [CreateEventSubscription](#) Reference AWS CLI .

## create-hsm-client-certificate

Il seguente esempio di codice mostra come utilizzare `create-hsm-client-certificate`.

### AWS CLI

Per creare un certificato HSM client

L'esempio seguente genera un certificato HSM client che un cluster può utilizzare per connettersi a un HSM.

```
aws redshift create-hsm-client-certificate \
  --hsm-client-certificate-identifier myhsmclientcert
```

Output:

```
{
  "HsmClientCertificate": {
    "HsmClientCertificateIdentifier": "myhsmclientcert",
    "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----
MIICiEXAMPLECQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAGTEXAMPLEEwDgYDVQQHEwDTZWFDGxLMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAwTC01BTSBDb25EXAMPLEIwEAYDVQQDEw1UZXR0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb251QGFTYXpvbi5jb2EXAMPLETEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCVVMxCzAJBgNVBEXAMPLEMRAwDgYD
EXAMPLETZWF0dGxLMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
b2x1MRIwEAEXAMPLEEw1UZXR0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb251QGFT
YXpvbi5jb20wgZ8wDQYJKEXAMPLEAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk6EXAMPLE3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugEXAMPLEzZswY6786m86gpE
Ibb30hjZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEEXAMPLEEAtCu4
nUhVvxYUEXAMPLEEh8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJI1J00zbhNYS5f6GEXAMPLE10ZxBHjJnyp3780D8uTs7fLvjx79LjStB
NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\n",
    "Tags": []
  }
}
```

Per ulteriori informazioni, consulta [Amazon Redshift API Permissions Reference](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [CreateHsmClientCertificate](#) Reference AWS CLI .

## create-hsm-configuration

Il seguente esempio di codice mostra come utilizzare `create-hsm-configuration`.

### AWS CLI

Per creare una HSM configurazione

L'esempio seguente crea la HSM configurazione specificata che contiene le informazioni richieste da un cluster per archiviare e utilizzare le chiavi di crittografia del database in un modulo di sicurezza hardware (HSM).

```
aws redshift create-hsm-configuration /
  --hsm-configuration-identifier 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
```

Output:

```
{
  "HsmConfiguration": {
    "HsmConfigurationIdentifier": "myhsmconnection",
    "Description": "My HSM connection",
    "HsmIpAddress": "192.0.2.09",
    "HsmPartitionName": "myhsmpartition",
    "Tags": []
  }
}
```

- Per API i dettagli, vedere [CreateHsmConfiguration](#) in AWS CLI Command Reference.

## create-snapshot-copy-grant

Il seguente esempio di codice mostra come utilizzare `create-snapshot-copy-grant`.

### AWS CLI

Per creare un'istantanea, concedi la copia

L'`create-snapshot-copy-grant` seguente crea una concessione di copia delle istantanee e crittografa le istantanee copiate in una regione di destinazione. AWS

```
aws redshift create-snapshot-copy-grant \  
  --snapshot-copy-grant-name mysnapshotcopygrantname
```

Output:

```
{  
  "SnapshotCopyGrant": {  
    "SnapshotCopyGrantName": "mysnapshotcopygrantname",  
    "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/  
bPxRfih3yCo8nvbEXAMPLEKEY",  
    "Tags": []  
  }  
}
```

Per ulteriori informazioni, consulta [Amazon Redshift Database Encryption](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateSnapshotCopyGrant](#)Reference.

## `create-snapshot-schedule`

Il seguente esempio di codice mostra come utilizzare `create-snapshot-schedule`.

AWS CLI

Per creare una pianificazione delle istantanee

L'`create-snapshot-schedule` seguente crea una pianificazione degli snapshot con la descrizione specificata e una frequenza ogni 12 ore.

```
aws redshift create-snapshot-schedule \  
  --schedule-definitions "rate(12 hours)" \  
  --schedule-identifier mysnapshotschedule \  
  --schedule-description "My schedule description"
```

Output:

```
{  
  "ScheduleDefinitions": [  

```

```
    "rate(12 hours)"
  ],
  "ScheduleIdentifier": "mysnapshotschedule",
  "ScheduleDescription": "My schedule description",
  "Tags": []
}
```

Per ulteriori informazioni, consulta [Automated Snapshot Schedules](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateSnapshotSchedule](#)Reference.

## create-tags

Il seguente esempio di codice mostra come utilizzare `create-tags`.

### AWS CLI

Per creare tag per un cluster

L'esempio seguente aggiunge la coppia chiave/valore del tag specificata al cluster specificato.

```
aws redshift create-tags \  
  --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \  
  --tags "Key"="mytags","Value"="tag1"
```

Il comando non produce output.

Per ulteriori informazioni, consulta [Tagging Resources in Amazon Redshift nella Amazon](#) Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [CreateTags](#)Reference AWS CLI .

## delete-cluster-parameter-group

Il seguente esempio di codice mostra come utilizzare `delete-cluster-parameter-group`.

### AWS CLI

L'esempio di eliminazione di un parametro del cluster elimina un parametro del cluster Group. Comando:

```
aws redshift delete-cluster-parameter-group --parameter-group-name
myclusterparametergroup
```

- Per API i dettagli, vedere [DeleteClusterParameterGroup](#) in AWS CLI Command Reference.

## **delete-cluster-security-group**

Il seguente esempio di codice mostra come utilizzare `delete-cluster-security-group`.

### AWS CLI

Eliminare un GroupThis esempio di Cluster Security elimina un gruppo di sicurezza del cluster. Comando:

```
aws redshift delete-cluster-security-group --cluster-security-group-name
mysecuritygroup
```

- Per API i dettagli, vedere [DeleteClusterSecurityGroup](#) in AWS CLI Command Reference.

## **delete-cluster-snapshot**

Il seguente esempio di codice mostra come utilizzare `delete-cluster-snapshot`.

### AWS CLI

L' SnapshotThis esempio di eliminazione di un cluster elimina uno snapshot. Comando:

```
aws redshift delete-cluster-snapshot --snapshot-identifier my-snapshot-id
```

- Per i API dettagli, vedere in Command Reference. [DeleteClusterSnapshot](#) AWS CLI

## **delete-cluster-subnet-group**

Il seguente esempio di codice mostra come utilizzare `delete-cluster-subnet-group`.

### AWS CLI

Eliminare un GroupThis esempio di sottorete Cluster elimina un gruppo di sottoreti del cluster. Comando:

```
aws redshift delete-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
```

Risultato:

```
{
  "ResponseMetadata": {
    "RequestId": "253fbffd-6993-11e2-bc3a-47431073908a"
  }
}
```

- Per i API dettagli, vedere in Command Reference. [DeleteClusterSubnetGroup](#)AWS CLI

## delete-cluster

Il seguente esempio di codice mostra come utilizzare `delete-cluster`.

AWS CLI

L' `SnapshotThis` esempio Elimina un cluster senza un cluster finale elimina un cluster, forzando l'eliminazione dei dati in modo che non venga creata alcuna istantanea finale del cluster.

Comando:

```
aws redshift delete-cluster --cluster-identifier mycluster --skip-final-cluster-snapshot
```

L' `SnapshotThis` esempio Elimina un cluster, `Allowing a Final Cluster` elimina un cluster, ma specifica uno snapshot finale del cluster. Comando:

```
aws redshift delete-cluster --cluster-identifier mycluster --final-cluster-snapshot-identifier myfinalsnapshot
```

- Per i API dettagli, vedere in Command Reference. [DeleteCluster](#)AWS CLI

## delete-event-subscription

Il seguente esempio di codice mostra come utilizzare `delete-event-subscription`.

AWS CLI

Per eliminare la sottoscrizione agli eventi

L'`delete-event-subscription` seguente elimina la sottoscrizione di notifica degli eventi specificata.

```
aws redshift delete-event-subscription \  
  --subscription-name mysubscription
```

Il comando non produce output.

Per ulteriori informazioni, consulta la sezione [Abbonamento alle notifiche degli eventi di Amazon Redshift](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [DeleteEventSubscription](#) Reference AWS CLI .

## **delete-hsm-client-certificate**

Il seguente esempio di codice mostra come utilizzare `delete-hsm-client-certificate`.

AWS CLI

Per eliminare il certificato HSM del client

L'`delete-hsm-client-certificate` seguente elimina un certificato HSM client.

```
aws redshift delete-hsm-client-certificate \  
  --hsm-client-certificate-identifier myhsmClientcert
```

Il comando non produce output.

Per ulteriori informazioni, consulta [Amazon Redshift API Permissions Reference](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [DeleteHsmClientCertificate](#) Reference AWS CLI .

## **delete-hsm-configuration**

Il seguente esempio di codice mostra come utilizzare `delete-hsm-configuration`.

AWS CLI

Per eliminare una HSM configurazione



L'`delete-hsm-configuration`esempio seguente elimina la HSM configurazione specificata dall' AWS account corrente.

```
aws redshift delete-hsm-configuration /  
--hsm-configuration-identifier myhsmconnection
```

Il comando non produce output.

- Per API i dettagli, vedere [DeleteHsmConfiguration](#)in AWS CLI Command Reference.

## **delete-scheduled-action**

Il seguente esempio di codice mostra come utilizzare `delete-scheduled-action`.

AWS CLI

Per eliminare un'azione pianificata

L'`delete-scheduled-action`esempio seguente elimina l'azione pianificata specificata.

```
aws redshift delete-scheduled-action \  
--scheduled-action-name myscheduledaction
```

Il comando non produce output.

- Per API i dettagli, vedere [DeleteScheduledAction](#)in AWS CLI Command Reference.

## **delete-snapshot-copy-grant**

Il seguente esempio di codice mostra come utilizzare `delete-snapshot-copy-grant`.

AWS CLI

Per eliminare un'istantanea, copia, concedi

L'`delete-snapshot-copy-grant`esempio seguente elimina la concessione di copia dell'istantanea specificata.

```
aws redshift delete-snapshot-copy-grant \  
--snapshot-copy-grant-name mysnapshotcopygrantname
```

Il comando non produce output.

Per ulteriori informazioni, consulta [Amazon Redshift Database Encryption](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteSnapshotCopyGrant](#)Reference.

## delete-snapshot-schedule

Il seguente esempio di codice mostra come utilizzare `delete-snapshot-schedule`.

### AWS CLI

Per eliminare la pianificazione delle istantanee

L'esempio seguente elimina la pianificazione degli snapshot specificata. È necessario dissociare i cluster prima di eliminare la pianificazione.

```
aws redshift delete-snapshot-schedule \  
  --schedule-identifier mysnapshotschedule
```

Il comando non produce output.

Per ulteriori informazioni, consulta [Automated Snapshot Schedules](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteSnapshotSchedule](#)Reference.

## delete-tags

Il seguente esempio di codice mostra come utilizzare `delete-tags`.

### AWS CLI

Per eliminare i tag da un cluster

L'esempio seguente elimina i tag con i nomi di chiave specificati dal cluster specificato.

```
aws redshift delete-tags \  
  --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \  
  --tag-keys "clustertagkey" "clustertagvalue"
```

Il comando non produce output.

Per ulteriori informazioni, consulta [Tagging Resources in Amazon Redshift nella Amazon Redshift Cluster Management Guide](#).

- Per API i dettagli, consulta Command [DeleteTags](#)Reference AWS CLI .

## **describe-account-attributes**

Il seguente esempio di codice mostra come utilizzare `describe-account-attributes`.

### AWS CLI

Per descrivere gli attributi di un AWS account

L'`describe-account-attributes`esempio seguente visualizza gli attributi associati all' AWS account chiamante.

```
aws redshift describe-account-attributes
```

Output:

```
{
  "AccountAttributes": [
    {
      "AttributeName": "max-defer-maintenance-duration",
      "AttributeValues": [
        {
          "AttributeValue": "45"
        }
      ]
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeAccountAttributes](#)in AWS CLI Command Reference.

## **describe-cluster-db-revisions**

Il seguente esempio di codice mostra come utilizzare `describe-cluster-db-revisions`.

### AWS CLI

Per descrivere le revisioni del database per un cluster

L' `describe-cluster-db-revisions` esempio seguente visualizza i dettagli di una matrice di `ClusterDbRevision` oggetti per il cluster specificato.

```
aws redshift describe-cluster-db-revisions \  
  --cluster-identifier mycluster
```

Output:

```
{  
  "ClusterDbRevisions": [  
    {  
      "ClusterIdentifier": "mycluster",  
      "CurrentDatabaseRevision": "11420",  
      "DatabaseRevisionReleaseDate": "2019-11-22T16:43:49.597Z",  
      "RevisionTargets": []  
    }  
  ]  
}
```

- Per API i dettagli, vedere [DescribeClusterDbRevisions](#) in AWS CLI Command Reference.

## describe-cluster-parameter-groups

Il seguente esempio di codice mostra come utilizzare `describe-cluster-parameter-groups`.

AWS CLI

L' `Groups` This esempio `Get a Description of All Cluster Parameter` restituisce una descrizione di tutti i gruppi di parametri del cluster per l'account, con le intestazioni delle colonne. Per impostazione predefinita, l'output è in JSON format. Comando:

```
aws redshift describe-cluster-parameter-groups
```

Risultato:

```
{  
  "ParameterGroups": [  
    {  
      "ParameterGroupFamily": "redshift-1.0",  
      "Description": "My first cluster parameter group",  
    }  
  ]  
}
```

```

    "ParameterGroupName": "myclusterparametergroup"
  } ],
  "ResponseMetadata": {
    "RequestId": "8ceb8f6f-64cc-11e2-bea9-49e0ce183f07"
  }
}

```

È inoltre possibile ottenere le stesse informazioni in formato testo utilizzando l'opzione. Comando:

```
--output text
```

```
--output textOpzione.Comando:
```

Opzione.comando:

```
aws redshift describe-cluster-parameter-groups --output text
```

Risultato:

```

redshift-1.0      My first cluster parameter group      myclusterparametergroup
RESPONSEMETADATA 9e665a36-64cc-11e2-8f7d-3b939af52818

```

- Per API i dettagli, vedere [DescribeClusterParameterGroups](#) in AWS CLI Command Reference.

## describe-cluster-parameters

Il seguente esempio di codice mostra come utilizzare `describe-cluster-parameters`.

AWS CLI

L' GroupThis esempio Recupera i parametri per un parametro di cluster specificato recupera i parametri per il gruppo di parametri denominato. Per impostazione predefinita, l'output è in JSON formato.Comando:

```
aws redshift describe-cluster-parameters --parameter-group-name
myclusterparametergroup
```

Risultato:

```
{
```

```

"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...)
]
}

```

È inoltre possibile ottenere le stesse informazioni in formato testo utilizzando l'opzione. Comando:

```
--output text
```

Opzione.comando:

```
aws redshift describe-cluster-parameters --parameter-group-name
myclusterparametergroup --output text
```

Risultato:

```

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

```

- Per API i dettagli, vedere [DescribeClusterParameters](#) in AWS CLI Command Reference.

## describe-cluster-security-groups

Il seguente esempio di codice mostra come utilizzare `describe-cluster-security-groups`.

### AWS CLI

L' `Groups` This esempio `Get a Description of All Cluster Security` restituisce una descrizione di tutti i gruppi di sicurezza del cluster per l'account. Per impostazione predefinita, l'output è in JSON formato. Comando:

```
aws redshift describe-cluster-security-groups
```

### Risultato:

```

{
  "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...)
]
}

```

- Per API i dettagli, vedere [DescribeClusterSecurityGroups](#) in AWS CLI Command Reference.

## describe-cluster-snapshots

Il seguente esempio di codice mostra come utilizzare `describe-cluster-snapshots`.

### AWS CLI

L' `describe-cluster-snapshots` restituisce una descrizione di tutte le istantanee del cluster per l'account. Per impostazione predefinita, l'output è in JSON formato. Comando:

```
aws redshift describe-cluster-snapshots
```

### Risultato:

```

{
  "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...)

```

- Per API i dettagli, vedere [DescribeClusterSnapshots](#) in AWS CLI Command Reference.

## describe-cluster-subnet-groups

Il seguente esempio di codice mostra come utilizzare `describe-cluster-subnet-groups`.

## AWS CLI

L' GroupsThis esempio Get a Description of All Cluster Subnet restituisce una descrizione di tutti i gruppi di sottoreti del cluster. Per impostazione predefinita, l'output è in JSON formato. Comando:

```
aws redshift describe-cluster-subnet-groups
```

Risultato:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [DescribeClusterSubnetGroups](#) in AWS CLI Command Reference.

## describe-cluster-tracks

Il seguente esempio di codice mostra come utilizzare `describe-cluster-tracks`.

## AWS CLI

Per descrivere le tracce del cluster

L'output di `aws redshift describe-cluster-tracks` seguente mostra i dettagli delle tracce di manutenzione disponibili.

```
aws redshift describe-cluster-tracks \
  --maintenance-track-name current
```

Output:

```
{
  "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"
            }
          ]
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione [Choose Cluster Maintenance Tracks](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta [DescribeClusterTracks AWS CLI Command Reference](#).

## describe-cluster-versions

Il seguente esempio di codice mostra come utilizzare `describe-cluster-versions`.

### AWS CLI

L' `describe-cluster-versions` restituisce una descrizione di tutte le versioni del cluster. Per impostazione predefinita, l'output è in JSON formato. Comando:

```
aws redshift describe-cluster-versions
```

Risultato:

```
{
  "ClusterVersions": [
    {
      "ClusterVersion": "1.0",
      "Description": "Initial release",
      "ClusterParameterGroupFamily": "redshift-1.0"
    }
  ],
  "ResponseMetadata": {
    "RequestId": "16a53de3-64cc-11e2-bec0-17624ad140dd"
  }
}
```

- Per API i dettagli, vedere [DescribeClusterVersions](#) in AWS CLI Command Reference.

## describe-clusters

Il seguente esempio di codice mostra come utilizzare `describe-clusters`.

### AWS CLI

L' `describe-clusters` restituisce una descrizione di tutti i cluster dell'account. Per impostazione predefinita, l'output è in JSON formato. Comando:

```
aws redshift describe-clusters
```

Risultato:

```

{
  "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"
  }
}

```

È inoltre possibile ottenere le stesse informazioni in formato testo utilizzando l'opzione. Comando:

```
--output text
```

```
--output text Opzione.Comando:
```

**Opzione.comando:**

```
aws redshift describe-clusters --output text
```

**Risultato:**

```
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
```

- Per API i dettagli, vedere [DescribeClusters](#) in AWS CLI Command Reference.

**describe-default-cluster-parameters**

Il seguente esempio di codice mostra come utilizzare `describe-default-cluster-parameters`.

**AWS CLI**

L' `Parameters` This esempio `Get a Description of Default Cluster` restituisce una descrizione dei parametri di cluster predefiniti per la `redshift-1.0` famiglia. Per impostazione predefinita, l'output è in JSON formato. Comando:

```
aws redshift describe-default-cluster-parameters --parameter-group-family
redshift-1.0
```

**Risultato:**

```
{
  "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...)
]
}
}

```

Per visualizzare un elenco di famiglie di gruppi di parametri valide, utilizzate il comando.

`describe-cluster-parameter-groups`

`describe-cluster-parameter-groups` comando.

comando.

- Per API i dettagli, vedere [DescribeDefaultClusterParameters](#) in AWS CLI Command Reference.

## **describe-event-categories**

Il seguente esempio di codice mostra come utilizzare `describe-event-categories`.

AWS CLI

Per descrivere le categorie di eventi per un cluster

L'esempio seguente visualizza i dettagli per le categorie di eventi per un cluster.

```
aws redshift describe-event-categories \
  --source-type cluster
```

Output:

```

{
  "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"
        }
      ]
    }
  ]
}

```

- Per API i dettagli, vedere [DescribeEventCategories](#) in AWS CLI Command Reference.



## describe-event-subscriptions

Il seguente esempio di codice mostra come utilizzare `describe-event-subscriptions`.

### AWS CLI

Per descrivere le sottoscrizioni agli eventi

L'esempio seguente visualizza le sottoscrizioni alle notifiche degli eventi per la sottoscrizione specificata.

```
aws redshift describe-event-subscriptions \
  --subscription-name mysubscription
```

Output:

```
{
  "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": []
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione [Abbonamento alle notifiche degli eventi di Amazon Redshift](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [DescribeEventSubscriptions](#) Reference AWS CLI .

## describe-events

Il seguente esempio di codice mostra come utilizzare `describe-events`.

## AWS CLI

Descrivi tutti gli eventi: questo esempio restituisce tutti gli eventi. Per impostazione predefinita, l'output è in JSON formato. Comando:

```
aws redshift describe-events
```

Risultato:

```
{
  "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"
  }
}
```

È inoltre possibile ottenere le stesse informazioni in formato testo utilizzando l'opzione. Comando:

```
--output text
```

--output text Opzione. Comando:

Opzione. comando:

```
aws redshift describe-events --output text
```

Risultato:

```
2013-01-22T19:17:03.640Z    myclusterparametergroup Cluster parameter group
myclusterparametergroup has been created.    cluster-parameter-group
RESPONSEMETADATA    8e5fe765-64c9-11e2-bce3-e56f52c50e17
```

- Per API i dettagli, vedere [DescribeEvents](#) in AWS CLI Command Reference.

## describe-hsm-client-certificates

Il seguente esempio di codice mostra come utilizzare `describe-hsm-client-certificates`.

### AWS CLI

Per descrivere i certificati HSM client

L'esempio seguente visualizza i dettagli per il certificato HSM client specificato.

```
aws redshift describe-hsm-client-certificates \
  --hsm-client-certificate-identifier myhsmclientcert
```

Output:

```
{
  "HsmClientCertificates": [
    {
      "HsmClientCertificateIdentifier": "myhsmclientcert",
      "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----\
EXAMPLECAfICCQD6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAEXAMPLERAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAwTC01BTSBDb25zEXAMPLEwEAYDVQQDEwLUZXN0Q21sYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGftYXpvbi5jb20wHhEXAMPLEDI1MjA0EXAMPLEN
EXAMPLE0MjA0NTIxWjCBiDELMAKGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGEXAMPLEQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTSBDb25z
b2x1MRIwEAYDVQQDEwLUZXN0Q21sEXAMPLEdBgkqhkiG9w0BCQEWEG5vb25lQGft
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIEEXAMPLEMaK0dn+a4GmWIWJ
21uUSfwfEvySwTC2XADZ4nB+BLygVIk60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY67EXAMPLEE
EXAMPLEZnzcvcQAaRHhd1QWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9EXAMPLE6q+auNKyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0Fkb
FFbjvSfpJI1J00zbhNYS5f6GuoEEXAMPLEBHjJnyp3780D8uTs7fLvJx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\n",
      "Tags": []
    }
  ]
}
```

Per ulteriori informazioni, consulta [Amazon Redshift API Permissions Reference](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [DescribeHsmClientCertificates](#)Reference AWS CLI .

## describe-hsm-configurations

Il seguente esempio di codice mostra come utilizzare `describe-hsm-configurations`.

### AWS CLI

Per descrivere le HSM configurazioni

L'`describe-hsm-configurations`esempio seguente mostra i dettagli delle HSM configurazioni disponibili per l'account chiamante AWS .

```
aws redshift describe-hsm-configurations /  
--hsm-configuration-identifier myhsmconnection
```

Output:

```
{  
  "HsmConfigurations": [  
    {  
      "HsmConfigurationIdentifier": "myhsmconnection",  
      "Description": "My HSM connection",  
      "HsmIpAddress": "192.0.2.09",  
      "HsmPartitionName": "myhsmpartition",  
      "Tags": []  
    }  
  ]  
}
```

- Per API i dettagli, vedere [DescribeHsmConfigurations](#)in AWS CLI Command Reference.

## describe-logging-status

Il seguente esempio di codice mostra come utilizzare `describe-logging-status`.

### AWS CLI

Per descrivere lo stato di registrazione di un cluster

L'`describe-logging-status`esempio seguente mostra se le informazioni, come le interrogazioni e i tentativi di connessione, vengono registrate per un cluster.

```
aws redshift describe-logging-status \  
  --cluster-identifier mycluster
```

Output:

```
{  
  "LoggingEnabled": false  
}
```

Per ulteriori informazioni, consulta [Database Audit Logging](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeLoggingStatus](#)Reference.

## describe-node-configuration-options

Il seguente esempio di codice mostra come utilizzare `describe-node-configuration-options`.

AWS CLI

Per descrivere le opzioni di configurazione dei nodi

L'esempio seguente visualizza le proprietà delle possibili configurazioni di nodi, come il tipo di nodo, il numero di nodi e l'utilizzo del disco per lo snapshot del cluster specificato.

```
aws redshift describe-node-configuration-options \  
  --action-type restore-cluster \  
  --snapshot-identifier rs:mycluster-2019-12-09-16-42-43
```

Output:

```
{  
  "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
  }
]
}
```

Per ulteriori informazioni, consulta la sezione [Acquisto di nodi riservati Amazon Redshift](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeNodeConfigurationOptions](#) Reference.

## describe-orderable-cluster-options

Il seguente esempio di codice mostra come utilizzare `describe-orderable-cluster-options`.

### AWS CLI

L'OptionsThis esempio Describing All Orderable Cluster restituisce le descrizioni di tutte le opzioni di cluster ordinabili. Per impostazione predefinita, l'output è in formato. Comando: JSON

```
aws redshift describe-orderable-cluster-options
```

### Risultato:

```
{
  "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"
  }
}

```

È inoltre possibile ottenere le stesse informazioni in formato testo utilizzando l'opzione. Comando:

```
--output text
```

--output text Opzione. Comando:

Opzione.comando:

```
aws redshift describe-orderable-cluster-options --output text
```

Risultato:

```

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
```

- Per API i dettagli, vedere [DescribeOrderableClusterOptions](#) in AWS CLI Command Reference.

## describe-reserved-node-offerings

Il seguente esempio di codice mostra come utilizzare `describe-reserved-node-offerings`.

### AWS CLI

L' `OfferingsThis` esempio Descrivi il nodo riservato mostra tutte le offerte di nodi riservati disponibili per `Purchase`. Command:

```
aws redshift describe-reserved-node-offerings
```

### Risultato:

```
{
  "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"
}
}

```

Se desideri acquistare un'offerta di nodi riservati, puoi chiamare `purchase-reserved-node-offering` utilizzando un nodo valido. `ReservedNodeOfferingId`

`purchase-reserved-node-offering` utilizzando un valido `ReservedNodeOfferingId`.

utilizzando un valido `ReservedNodeOfferingId`.

`ReservedNodeOfferingId`.

.

- Per API i dettagli, vedere [DescribeReservedNodeOfferings](#) in AWS CLI Command Reference.

## describe-reserved-nodes

Il seguente esempio di codice mostra come utilizzare `describe-reserved-nodes`.

### AWS CLI

Descrivi l' NodesThis esempio riservato mostra un'offerta di nodi riservati che è stata acquistata. Comando:

```
aws redshift describe-reserved-nodes
```

Risultato:

```

{
  "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"
      }
    ]
  }
}

```

- Per i API dettagli, vedere [DescribeReservedNodes](#) in Command Reference.AWS CLI

## describe-resize

Il seguente esempio di codice mostra come utilizzare `describe-resize`.

### AWS CLI

Descrivi l' `ResizeThis` esempio che descrive l'ultimo ridimensionamento di un cluster. La richiesta riguardava 3 nodi di tipo `dw.hs1.8xlarge`. Command:

```
aws redshift describe-resize --cluster-identifier mycluster
```

### Risultato:

```

{
  "Status": "NONE",
  "TargetClusterType": "multi-node",
  "TargetNodeType": "dw.hs1.8xlarge",
  "ResponseMetadata": {

```

```
    "RequestId": "9f52b0b4-7733-11e2-aa9b-318b2909bd27"
  },
  "TargetNumberOfNodes": "3"
}
```

- Per API i dettagli, vedere [DescribeResize](#) in AWS CLI Command Reference.

## describe-scheduled-actions

Il seguente esempio di codice mostra come utilizzare `describe-scheduled-actions`.

### AWS CLI

Per descrivere le azioni pianificate

L'esempio seguente mostra i dettagli di tutte le azioni attualmente pianificate.

```
aws redshift describe-scheduled-actions
```

Output:

```
{
  "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"
      ]
    }
  ]
}
```

- Per API i dettagli, vedere [DescribeScheduledActions](#) in AWS CLI Command Reference.

## describe-snapshot-copy-grants

Il seguente esempio di codice mostra come utilizzare `describe-snapshot-copy-grants`.

### AWS CLI

Per descrivere le concessioni di copia delle istantanee

L'`describe-snapshot-copy-grant`sempio seguente visualizza i dettagli per la concessione di copia delle istantanee del cluster specificata.

```
aws redshift describe-snapshot-copy-grants \  
  --snapshot-copy-grant-name mynapshotcopygrantname
```

Output:

```
{  
  "SnapshotCopyGrants": [  
    {  
      "SnapshotCopyGrantName": "mynapshotcopygrantname",  
      "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/  
bPxRfih3yCo8nvbEXAMPLEKEY",  
      "Tags": []  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Amazon Redshift Database Encryption](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeSnapshotCopyGrants](#)Reference.

## describe-snapshot-schedules

Il seguente esempio di codice mostra come utilizzare `describe-snapshot-schedules`.

### AWS CLI

Per descrivere le pianificazioni delle istantanee

L'output del comando `aws redshift describe-snapshot-schedules` seguente visualizza i dettagli per la pianificazione delle istantanee del cluster specificata.

```
aws redshift describe-snapshot-schedules \
  --cluster-identifier mycluster \
  --schedule-identifier mynapshotschedule
```

Output:

```
{
  "SnapshotSchedules": [
    {
      "ScheduleDefinitions": [
        "rate(12 hours)"
      ],
      "ScheduleIdentifier": "mynapshotschedule",
      "ScheduleDescription": "My schedule description",
      "Tags": [],
      "AssociatedClusterCount": 1,
      "AssociatedClusters": [
        {
          "ClusterIdentifier": "mycluster",
          "ScheduleAssociationState": "ACTIVE"
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, consulta [Automated Snapshot Schedules](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeSnapshotSchedules](#) Reference.

## describe-storage

Il seguente esempio di codice mostra come utilizzare `describe-storage`.

AWS CLI

Per descrivere l'archiviazione

L'`describe-storage` seguente visualizza i dettagli sullo storage di backup e sulle dimensioni di archiviazione provvisorie per l'account.

```
aws redshift describe-storage
```

Output:

```
{
  "TotalBackupSizeInMegaBytes": 193149.0,
  "TotalProvisionedStorageInMegaBytes": 655360.0
}
```

Per ulteriori informazioni, consulta [Managing Snapshot Storage](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta AWS CLI Command [DescribeStorage](#) Reference.

## `describe-table-restore-status`

Il seguente esempio di codice mostra come utilizzare `describe-table-restore-status`.

AWS CLI

Per descrivere lo stato delle richieste di ripristino della tabella da uno snapshot del cluster

L'`describe-table-restore-status` seguente visualizza i dettagli delle richieste di ripristino delle tabelle effettuate per il cluster specificato.

```
aws redshift describe-table-restore-status /
  --cluster-identifier mycluster
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Restoring a table from a snapshot](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [DescribeTableRestoreStatus](#)Reference AWS CLI .

## describe-tags

Il seguente esempio di codice mostra come utilizzare `describe-tags`.

### AWS CLI

Per descrivere i tag

L'`describe-tags`esempio seguente visualizza le risorse associate dal cluster specificato ai nomi e ai valori dei tag specificati.

```

aws redshift describe-tags \
  --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \
  --tag-keys clustertagkey \
  --tag-values clustertagvalue

```

Output:

```

{
  "TaggedResources": [
    {
      "Tag": {
        "Key": "clustertagkey",
        "Value": "clustertagvalue"
      },
      "ResourceName": "arn:aws:redshift:us-
west-2:123456789012:cluster:mycluster",
      "ResourceType": "cluster"
    }
  ]
}

```

```
]
}
```

Per ulteriori informazioni, consulta [Tagging Resources in Amazon Redshift nella Amazon Redshift Cluster Management Guide](#).

- Per API i dettagli, consulta Command [DescribeTags](#)Reference AWS CLI .

## disable-snapshot-copy

Il seguente esempio di codice mostra come utilizzare `disable-snapshot-copy`.

### AWS CLI

Per disabilitare la copia delle istantanee per un cluster

L'esempio seguente disabilita la copia automatica di un'istananea per il cluster specificato.

```
aws redshift disable-snapshot-copy \
  --cluster-identifier mycluster
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, [consulta Copiare gli snapshot in un'altra AWS regione nella Amazon Redshift Cluster Management Guide](#).

- Per API i dettagli, consulta Command [DisableSnapshotCopy](#) Reference AWS CLI .

## enable-snapshot-copy

Il seguente esempio di codice mostra come utilizzare `enable-snapshot-copy`.

### AWS CLI

Per abilitare la copia delle istantanee per un cluster

L'esempio seguente abilita la copia automatica di un'istanza per il cluster specificato.

```
aws redshift enable-snapshot-copy \  
  --cluster-identifier mycluster \  
  --destination-region us-west-1
```

Output:

```
{  
  "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"
```

```
}  
}
```

Per ulteriori informazioni, [consulta Copiare gli snapshot in un'altra AWS regione nella Amazon Redshift Cluster Management Guide](#).

- Per API i dettagli, consulta Command [EnableSnapshotCopy](#)Reference AWS CLI .

## get-cluster-credentials

Il seguente esempio di codice mostra come utilizzare `get-cluster-credentials`.

### AWS CLI

Per ottenere le credenziali del cluster per un account AWS

L'`get-cluster-credentials` esempio seguente recupera le credenziali temporanee che consentono l'accesso a un database Amazon Redshift.

```
aws redshift get-cluster-credentials \  
  --db-user adminuser --db-name dev \  
  --cluster-identifier mycluster
```

Output:

```
{  
  "DbUser": "IAM:adminuser",  
  "DbPassword": "AMAFUyyuros/QjxPTtgzcsuQsqzIasdzJEN04aCtWDzXx109d6UmpkBtvEeqFly/  
EXAMPLE==",  
  "Expiration": "2019-12-10T17:25:05.770Z"  
}
```

Per ulteriori informazioni, consulta [Generazione delle credenziali del IAM database utilizzando Amazon CLI Redshift API o nella Amazon Redshift Cluster Management Guide](#).

- Per API i dettagli, consulta Command [GetClusterCredentials](#)Reference AWS CLI .

## get-reserved-node-exchange-offerings

Il seguente esempio di codice mostra come utilizzare `get-reserved-node-exchange-offerings`.

## AWS CLI

Per ottenere offerte di scambio di nodi riservati

L'get-reserved-node-exchange-offeringsesempio seguente recupera una matrice DC2 ReservedNodeOfferings che corrisponde al nodo DC1 riservato specificato.

```
aws redshift get-reserved-node-exchange-offerings \  
--reserved-node-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Upgrade Reserved Nodes With the AWS CLI](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [GetReservedNodeExchangeOfferings](#)Reference AWS CLI .

## modify-cluster-iam-roles

Il seguente esempio di codice mostra come utilizzaremodify-cluster-iam-roles.

## AWS CLI

Per modificare il IAM ruolo di un cluster

L'`modify-cluster-iam-roles` seguente rimuove il AWS IAM ruolo specificato dal cluster specificato.

```
aws redshift modify-cluster-iam-roles \  
  --cluster-identifier mycluster \  
  --remove-iam-roles arn:aws:iam::123456789012:role/myRedshiftRole
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Using Identity-Based Policies \(IAM Policies\) per Amazon Redshift nella Amazon Redshift Cluster Management Guide](#).

- Per i API dettagli, consulta Command Reference. [ModifyClusterIamRoles](#) AWS CLI

## modify-cluster-maintenance

Il seguente esempio di codice mostra come utilizzare `modify-cluster-maintenance`.

## AWS CLI

Per modificare la manutenzione del cluster

L'`modify-cluster-maintenance` seguente posticipa di 30 giorni la manutenzione del cluster specificato.

```
aws redshift modify-cluster-maintenance \  
  --cluster-identifier mycluster \  
  --defer-maintenance \  
  --defer-maintenance-duration 30
```

Output:

```
{  
  "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-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-mUdVIFcT1B4SGhw6fyF",
    "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"
}
}
```

Per ulteriori informazioni, consulta [Cluster Maintenance](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta [ModifyClusterMaintenance AWS CLI Command Reference](#).

## modify-cluster-parameter-group

Il seguente esempio di codice mostra come utilizzare `modify-cluster-parameter-group`.

### AWS CLI

Modificare un parametro in un gruppo di parametri

L'esempio seguente modifica il parametro `wlm_json_configuration` per la gestione del carico di lavoro. Accetta i parametri da un file che contiene i contenuti mostrati di seguito. JSON

```
aws redshift modify-cluster-parameter-group \  
  --parameter-group-name myclusterparametergroup \  
  --parameters file://modify_pg.json
```

Contenuto di `modify_pg.json`.

```
[  
  {  
    "ParameterName": "wlm_json_configuration",  
    "ParameterValue": "[{\\"user_group\\":\\"example_user_group1\\",\\"query_group\\":  
  \\"example_query_group1\\", \\"query_concurrency\\":7},{\\"query_concurrency\\":5}]"  
  }  
]
```

Output:

```
{  
  "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"  
  }  
}
```

- Per API i dettagli, vedere [ModifyClusterParameterGroup](#) in AWS CLI Command Reference.

## modify-cluster-snapshot-schedule

Il seguente esempio di codice mostra come utilizzare `modify-cluster-snapshot-schedule`.

## AWS CLI

Per modificare la pianificazione delle istantanee del cluster

L'`modify-cluster-snapshot-schedule` seguente rimuove la pianificazione delle istantanee specificata dal cluster specificato.

```
aws redshift modify-cluster-snapshot-schedule \  
  --cluster-identifier mycluster \  
  --schedule-identifier mysnapshotschedule \  
  --disassociate-schedule
```

Il comando non produce output.

Per ulteriori informazioni, consulta [Automated Snapshot Schedules](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta AWS CLI Command [ModifyClusterSnapshotSchedule](#) Reference.

## `modify-cluster-snapshot`

Il seguente esempio di codice mostra come utilizzare `modify-cluster-snapshot`.

### AWS CLI

Per modificare l'istantanea del cluster

L'`modify-cluster-snapshot` seguente imposta l'impostazione del periodo di conservazione manuale per lo snapshot del cluster specificato su un valore di 10 giorni.

```
aws redshift modify-cluster-snapshot \  
  --snapshot-identifier mycluster-2019-11-06-16-32 \  
  --manual-snapshot-retention-period 10
```

Output:

```
{  
  "Snapshot": {  
    "SnapshotIdentifier": "mycluster-2019-11-06-16-32",  
    "ClusterIdentifier": "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"
  }
}

```

Per ulteriori informazioni, consulta [Amazon Redshift Snapshots](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [ModifyClusterSnapshot](#)Reference AWS CLI .

## modify-cluster-subnet-group

Il seguente esempio di codice mostra come utilizzare `modify-cluster-subnet-group`.

## AWS CLI

L' GroupThis esempio di modifica delle sottoreti in una sottorete del cluster mostra come modificare l'elenco delle sottoreti in un gruppo di sottoreti della cache. Per impostazione predefinita, l'output è in formato.Comando: JSON

```
aws redshift modify-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
--subnet-ids subnet-763fdd1 subnet-ac830e9
```

Risultato:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [ModifyClusterSubnetGroup](#) in AWS CLI Command Reference.

## modify-cluster

Il seguente esempio di codice mostra come utilizzare `modify-cluster`.

## AWS CLI

Associare un gruppo di sicurezza a un Cluster. Questo esempio mostra come associare un gruppo di sicurezza del cluster al Cluster. Comando specificato:

```
aws redshift modify-cluster --cluster-identifier mycluster --cluster-security-groups mysecuritygroup
```

Modifica la finestra di manutenzione per Cluster. Questo mostra come modificare la finestra di manutenzione settimanale preferita per un cluster in modo che diventi la finestra di manutenzione minima di quattro ore che inizia la domenica alle 23:15 e termina il lunedì alle 3:15. Comando:

```
aws redshift modify-cluster --cluster-identifier mycluster --preferred-maintenance-window Sun:23:15-Mon:03:15
```

Modifica la password principale: nell' Cluster. Questo esempio viene illustrato come modificare la password principale per un Cluster. Comando:

```
aws redshift modify-cluster --cluster-identifier mycluster --master-user-password A1b2c3d4
```

- Per API i dettagli, vedere [ModifyCluster](#) in AWS CLI Command Reference.

## modify-event-subscription

Il seguente esempio di codice mostra come utilizzare `modify-event-subscription`.

### AWS CLI

Per modificare la sottoscrizione agli eventi

L'esempio seguente disattiva la sottoscrizione di notifica degli eventi specificata.

```
aws redshift modify-event-subscription \  
  --subscription-name mysubscription \  
  --no-enabled
```

Output:

```
{
```

```

"EventSubscription": {
  "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": false,
  "Tags": []
}
}

```

Per ulteriori informazioni, consulta la sezione [Abbonamento alle notifiche degli eventi di Amazon Redshift](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [ModifyEventSubscription](#) Reference AWS CLI .

## modify-scheduled-action

Il seguente esempio di codice mostra come utilizzare `modify-scheduled-action`.

### AWS CLI

Per modificare l'azione pianificata

L'`modify-scheduled-action` esempio seguente aggiunge una descrizione all'azione pianificata esistente specificata.

```

aws redshift modify-scheduled-action \
  --scheduled-action-name myscheduledaction \
  --scheduled-action-description "My scheduled action"

```

Output:

```

{
  "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"
]
}
```

- Per API i dettagli, vedere [ModifyScheduledAction](#) in AWS CLI Command Reference.

## modify-snapshot-copy-retention-period

Il seguente esempio di codice mostra come utilizzare `modify-snapshot-copy-retention-period`.

### AWS CLI

Per modificare il periodo di conservazione delle copie istantanee

L'`modify-snapshot-copy-retention-period` seguente modifica il numero di giorni in cui conservare le istantanee per il cluster specificato nella regione di destinazione dopo che sono state copiate dalla AWS regione di origine. AWS

```
aws redshift modify-snapshot-copy-retention-period \
  --cluster-identifier mycluster \
  --retention-period 15
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Snapshot Schedule Format](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta AWS CLI Command [ModifySnapshotCopyRetentionPeriod](#) Reference.

## modify-snapshot-schedule

Il seguente esempio di codice mostra come utilizzare `modify-snapshot-schedule`.

AWS CLI

Per modificare la pianificazione delle istantanee

L'`modify-snapshot-schedule` esempio seguente modifica la frequenza della pianificazione delle istantanee specificata ogni 10 ore.

```

aws redshift modify-snapshot-schedule \
  --schedule-identifier mynapshotschedule \
  --schedule-definitions "rate(10 hours)"

```

Output:

```

{
  "ScheduleDefinitions": [
    "rate(10 hours)"
  ],

```

```

    "ScheduleIdentifier": "mysnapshotschedule",
    "ScheduleDescription": "My schedule description",
    "Tags": []
  }

```

Per ulteriori informazioni, consulta [Snapshot Schedule Format](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta AWS CLI Command [ModifySnapshotScheduleReference](#).

## purchase-reserved-node-offering

Il seguente esempio di codice mostra come utilizzare `purchase-reserved-node-offering`.

### AWS CLI

L' `NodeThis` esempio di acquisto di un nodo riservato mostra come acquistare un'offerta su un nodo riservato. `reserved-node-offering-id` Si ottiene chiamando `describe-reserved-node-offerings`. Command:

```
aws redshift purchase-reserved-node-offering --reserved-node-offering-id ceb6a579-cf4c-4343-be8b-d832c45ab51c
```

### Risultato:

```

{
  "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"
  }
}
```

- Per API i dettagli, vedere [PurchaseReservedNodeOffering](#) in AWS CLI Command Reference.

## reboot-cluster

Il seguente esempio di codice mostra come utilizzare `reboot-cluster`.

### AWS CLI

Riavvia un Cluster. Questo esempio riavvia un cluster. Per impostazione predefinita, l'output è in JSON formato. Comando:

```
aws redshift reboot-cluster --cluster-identifier mycluster
```

### Risultato:

```
{
  "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"
}
}

```

- Per API i dettagli, vedere [RebootCluster](#) in AWS CLI Command Reference.

## reset-cluster-parameter-group

Il seguente esempio di codice mostra come utilizzare `reset-cluster-parameter-group`.

### AWS CLI

L' `GroupThis` esempio di `reset Parameters in a Parameter` mostra come reimpostare tutti i parametri in un gruppo di parametri. Comando:

```
aws redshift reset-cluster-parameter-group --parameter-group-name
myclusterparametergroup --reset-all-parameters
```

- Per API i dettagli, vedere [ResetClusterParameterGroup](#) in AWS CLI Command Reference.

## resize-cluster

Il seguente esempio di codice mostra come utilizzare `resize-cluster`.

### AWS CLI

Per ridimensionare il cluster

L'`resize-cluster` seguente ridimensiona il cluster specificato.

```
aws redshift resize-cluster \  
  --cluster-identifier mycluster \  
  --cluster-type multi-node \  
  --node-type dc2.large \  
  --number-of-nodes 6 \  
  --classic
```

Output:

```
{  
  "Cluster": {  
    "ClusterIdentifier": "mycluster",  
    "NodeType": "dc2.large",  
    "ClusterStatus": "resizing",  
    "ClusterAvailabilityStatus": "Modifying",  
    "MasterUsername": "adminuser",  
    "DBName": "dev",  
    "Endpoint": {  
      "Address": "mycluster.cmeaswquae.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
}
}
```

Per ulteriori informazioni, consulta [Resizing a Cluster nella Amazon Redshift Cluster Management Guide](#).

- Per API i dettagli, consulta Command [ResizeCluster](#)Reference AWS CLI .

## restore-from-cluster-snapshot

Il seguente esempio di codice mostra come utilizzare `restore-from-cluster-snapshot`.

### AWS CLI

Ripristina un cluster Da un Snapshot This esempio ripristina un cluster da uno snapshot. Comando:

```
aws redshift restore-from-cluster-snapshot --cluster-identifier mycluster-clone --
snapshot-identifier my-snapshot-id
```

### Risultato:

```
{
  "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"
  }
}
```



```
}  
}
```

- Per i API dettagli, vedere in Command Reference. [RestoreFromClusterSnapshot](#) AWS CLI

## restore-table-from-cluster-snapshot

Il seguente esempio di codice mostra come utilizzare `restore-table-from-cluster-snapshot`.

### AWS CLI

Per ripristinare la tabella da un'istantanea del cluster

L'esempio seguente crea una nuova tabella dalla tabella specificata nello snapshot del cluster specificato.

```
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
```

Output:

```
{  
  "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"  
  }  
}
```

```
}
```

Per ulteriori informazioni, consulta [Restoring a table from a snapshot](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta Command [RestoreTableFromClusterSnapshot](#)Reference AWS CLI .

## **revoke-cluster-security-group-ingress**

Il seguente esempio di codice mostra come utilizzare `revoke-cluster-security-group-ingress`.

### AWS CLI

L' GroupThis esempio Revoke Access from an EC2 Security revoca l'accesso a un gruppo di sicurezza Amazon denominato. Comando: EC2

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name
mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-
owner-id 123445677890
```

La revoca dell'accesso a un esempio revoca l'accesso a un intervallo. Comando: CIDR rangeThis CIDR

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name
mysecuritygroup --cidrip 192.168.100.100/32
```

- Per i API dettagli, vedere in Command Reference. [RevokeClusterSecurityGroupIngress](#)AWS CLI

## **revoke-snapshot-access**

Il seguente esempio di codice mostra come utilizzare `revoke-snapshot-access`.

### AWS CLI

Revoca l'autorizzazione di un AWS account al ripristino Un SnapshotThis esempio revoca l'autorizzazione dell' AWS account 444455556666 a ripristinare l'istantanea. `my-snapshot-id` Per impostazione predefinita, l'output è in formato. Comando: JSON

```
aws redshift revoke-snapshot-access --snapshot-id my-snapshot-id --account-with-restore-access 444455556666
```

Risultato:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [RevokeSnapshotAccess](#) in AWS CLI Command Reference.

## rotate-encryption-key

Il seguente esempio di codice mostra come utilizzare `rotate-encryption-key`.

### AWS CLI

Per ruotare la chiave di crittografia per un cluster

L'rotate-encryption-keyesempio seguente ruota la chiave di crittografia per il cluster specificato.

```
aws redshift rotate-encryption-key \  
--cluster-identifier mycluster
```

Output:

```
{  
  "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/
bPxRfih3yCo8nvbEXAMPLEKEY",
    "EnhancedVpcRouting": false,
    "IamRoles": [
      {
        "IamRoleArn": "arn:aws:iam::123456789012:role/myRedshiftRole",
        "ApplyStatus": "in-sync"
      }
    ],
    "MaintenanceTrackName": "current",
    "DeferredMaintenanceWindows": [],
    "NextMaintenanceWindowStartTime": "2019-12-14T16:00:00Z"
  }
}
```

Per ulteriori informazioni, consulta [Amazon Redshift Database Encryption](#) nella Amazon Redshift Cluster Management Guide.

- Per API i dettagli, consulta AWS CLI Command [RotateEncryptionKey](#)Reference.

## Esempi di utilizzo di Amazon Rekognition AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon AWS Command Line Interface Rekognition.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### compare-faces

Il seguente esempio di codice mostra come utilizzare `compare-faces`.

Per ulteriori informazioni, consulta [Confronto dei volti nelle immagini](#).

### AWS CLI

Per confrontare i volti in due immagini

Il `compare-faces` comando seguente confronta i volti in due immagini archiviate in un bucket Amazon S3.

```
aws rekognition compare-faces \  
  --source-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"source.jpg"}}' \  
  --target-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"target.jpg"}}'
```

Output:

```
{  
  "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
}
}
```

Per ulteriori informazioni, consulta [Comparazione dei volti nelle immagini nella Amazon Rekognition Developer Guide](#).

- Per API i dettagli, consulta Command [CompareFaces](#)Reference AWS CLI .

## **create-collection**

Il seguente esempio di codice mostra come utilizzare `create-collection`.

Per ulteriori informazioni, consulta [Creazione di una raccolta](#).

### AWS CLI

Per creare una collezione

Il `create-collection` comando seguente crea una raccolta con il nome specificato.

```
aws rekognition create-collection \  
  --collection-id "MyCollection"
```

Output:

```
{  
  "CollectionArn": "aws:rekognition:us-west-2:123456789012:collection/  
MyCollection",  
  "FaceModelVersion": "4.0",  
  "StatusCode": 200  
}
```

Per ulteriori informazioni, consulta [Creating a Collection](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [CreateCollection](#)Reference AWS CLI .

## **create-stream-processor**

Il seguente esempio di codice mostra come utilizzare `create-stream-processor`.

### AWS CLI

Per creare un nuovo stream processor

L'esempio seguente crea un nuovo stream processor con la configurazione specificata.



```
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}}'
```

Output:

```
{  
  "StreamProcessorArn": "arn:aws:rekognition:us-  
west-2:123456789012:streamprocessor/my-stream-processor"  
}
```

Per ulteriori informazioni, consulta [Lavorare con i video in streaming](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [CreateStreamProcessor](#) Reference AWS CLI .

## delete-collection

Il seguente esempio di codice mostra come utilizzare `delete-collection`.

Per ulteriori informazioni, consulta [Eliminazione di una raccolta](#).

AWS CLI

Per eliminare una raccolta

Il `delete-collection` comando seguente elimina la raccolta specificata.

```
aws rekognition delete-collection \  
  --collection-id MyCollection
```

Output:

```
{  
  "StatusCode": 200  
}
```

Per ulteriori informazioni, consulta [Eliminazione di una raccolta](#) nella Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta [DeleteCollection](#) Command Reference.AWS CLI

## **delete-faces**

Il seguente esempio di codice mostra come utilizzare `delete-faces`.

Per ulteriori informazioni, consulta [Eliminazione dei volti da una raccolta](#).

### AWS CLI

Per eliminare volti da una raccolta

Il `delete-faces` comando seguente elimina la faccia specificata da una raccolta.

```
aws rekognition delete-faces \  
  --collection-id MyCollection \  
  --face-ids '["0040279c-0178-436e-b70a-e61b074e96b0"]'
```

Output:

```
{  
  "DeletedFaces": [  
    "0040279c-0178-436e-b70a-e61b074e96b0"  
  ]  
}
```

Per ulteriori informazioni, [consulta Eliminazione di volti da una raccolta](#) nella Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta [DeleteFaces](#) Command Reference.AWS CLI

## **delete-stream-processor**

Il seguente esempio di codice mostra come utilizzare `delete-stream-processor`.

### AWS CLI

Per eliminare uno stream processor

Il `delete-stream-processor` comando seguente elimina lo stream processor specificato.

```
aws rekognition delete-stream-processor \  
  --name my-stream-processor
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lavorare con i video in streaming](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [DeleteStreamProcessor](#)Reference AWS CLI .

## describe-collection

Il seguente esempio di codice mostra come utilizzare `describe-collection`.

Per ulteriori informazioni, consulta [Descrizione di una raccolta](#).

AWS CLI

Per descrivere una collezione

L'`describe-collection`esempio seguente visualizza i dettagli sulla raccolta specificata.

```
aws rekognition describe-collection \  
  --collection-id MyCollection
```

Output:

```
{  
  "FaceCount": 200,  
  "CreationTimestamp": 1569444828.274,  
  "CollectionARN": "arn:aws:rekognition:us-west-2:123456789012:collection/  
MyCollection",  
  "FaceModelVersion": "4.0"  
}
```

Per ulteriori informazioni, consulta [Descrivere una collezione](#) nella Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta [DescribeCollection](#)Command Reference.AWS CLI

## describe-stream-processor

Il seguente esempio di codice mostra come utilizzare `describe-stream-processor`.

### AWS CLI

Per ottenere informazioni su uno stream processor

Il `describe-stream-processor` comando seguente visualizza i dettagli sullo stream processor specificato.

```
aws rekognition describe-stream-processor \  
  --name my-stream-processor
```

Output:

```
{  
  "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  
}
```

Per ulteriori informazioni, consulta [Lavorare con i video in streaming](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [DescribeStreamProcessor](#)Reference AWS CLI .

## detect-faces

Il seguente esempio di codice mostra come utilizzare `detect-faces`.

Per ulteriori informazioni, consulta [Rilevamento dei volti in un'immagine](#).

### AWS CLI

Per rilevare i volti in un'immagine

Il `detect-faces` comando seguente rileva i volti nell'immagine specificata archiviata in un bucket Amazon S3.

```
aws rekognition detect-faces \  
  --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"MyFriend.jpg"}}' \  
  --attributes "ALL"
```

Output:

```
{  
  "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
      }
    ]
  }
```

Per ulteriori informazioni, consulta [Detecting Faces in an Image nella Amazon Rekognition Developer Guide](#).

- Per i API dettagli, consulta [DetectFaces](#) Command Reference.AWS CLI

## detect-labels

Il seguente esempio di codice mostra come utilizzare `detect-labels`.

Per ulteriori informazioni, consulta [Rilevamento delle etichette in un'immagine](#).

### AWS CLI

Per rilevare un'etichetta in un'immagine

L'`detect-labels` esempio seguente rileva scene e oggetti in un'immagine archiviata in un bucket Amazon S3.

```
aws rekognition detect-labels \
  --image '{"S3Object":{"Bucket":"bucket","Name":"image"}}'
```

Output:

```
{
  "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"
}

```

Per ulteriori informazioni, consulta [Detecting Labels in an Image nella Amazon Rekognition Developer Guide](#).

- Per i API dettagli, consulta [DetectLabels](#) Command Reference.AWS CLI

## detect-moderation-labels

Il seguente esempio di codice mostra come utilizzare `detect-moderation-labels`.

Per ulteriori informazioni, consulta [Rilevamento di immagini non appropriate](#).

### AWS CLI

Per rilevare contenuti non sicuri in un'immagine

Il `detect-moderation-labels` comando seguente rileva contenuti non sicuri nell'immagine specificata archiviata in un bucket Amazon S3.

```

aws rekognition detect-moderation-labels \
  --image "S3Object={Bucket=MyImageS3Bucket,Name=gun.jpg}"

```

Output:

```

{
  "ModerationModelVersion": "3.0",
  "ModerationLabels": [
    {
      "Confidence": 97.29618072509766,
      "ParentName": "Violence",
      "Name": "Weapon Violence"
    }
  ],
}

```



```
    {
      "Confidence": 97.29618072509766,
      "ParentName": "",
      "Name": "Violence"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Rilevamento di immagini non sicure](#) nella Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta Command Reference. [DetectModerationLabels](#) AWS CLI

## detect-text

Il seguente esempio di codice mostra come utilizzare `detect-text`.

Per ulteriori informazioni, consulta [Rilevamento del testo in un'immagine](#).

### AWS CLI

Per rilevare il testo in un'immagine

Il `detect-text` comando seguente rileva il testo nell'immagine specificata.

```
aws rekognition detect-text \
  --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"ExamplePicture.jpg"}}'
```

Output:

```
{
  "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
}
]
}

```

- Per API i dettagli, vedere [DetectText](#) in AWS CLI Command Reference.

## disassociate-faces

Il seguente esempio di codice mostra come utilizzare `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
```

- Per API i dettagli, vedere [DisassociateFaces](#) in AWS CLI Command Reference.

## get-celebrity-info

Il seguente esempio di codice mostra come utilizzare `get-celebrity-info`.

## AWS CLI

Per ottenere informazioni su una celebrità

Il `get-celebrity-info` comando seguente visualizza informazioni sulla celebrità specificata. Il `id` parametro proviene da una precedente chiamata `arecognize-celebrities`.

```
aws rekognition get-celebrity-info --id nnnnnnnn
```

Output:

```
{
  "Name": "Celeb A",
  "Urls": [
    "www.imdb.com/name/aaaaaaaaa"
  ]
}
```

Per ulteriori informazioni, consulta [Ottenerne informazioni su una celebrità](#) nella Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta [GetCelebrityInfo](#) Command Reference.AWS CLI

## get-celebrity-recognition

Il seguente esempio di codice mostra come utilizzare `get-celebrity-recognition`.

## AWS CLI

Per ottenere i risultati di un'operazione di riconoscimento di celebrità

Il `get-celebrity-recognition` comando seguente visualizza i risultati di un'operazione di riconoscimento di celebrità avviata in precedenza chiamando `start-celebrity-recognition`

```
aws rekognition get-celebrity-recognition \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Output:

```
{
```

```
"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/1111111111"
    ],
    "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
}
}
}

```

Per ulteriori informazioni, consulta [Riconoscere le celebrità in un video archiviato nella](#) Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta Command Reference. [GetCelebrityRecognition](#) AWS CLI

## get-content-moderation

Il seguente esempio di codice mostra come utilizzare `get-content-moderation`.

### AWS CLI

Per ottenere i risultati di un'operazione sui contenuti non sicura

Il `get-content-moderation` comando seguente visualizza i risultati di un'operazione relativa ai contenuti non sicuri avviata in precedenza mediante una chiamata `start-content-moderation`

```
aws rekognition get-content-moderation \  
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

Output:

```
{  
  "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  
  }  
}
```

Per ulteriori informazioni, consulta [Rilevamento di video archiviati non sicuri](#) nella Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta Command Reference. [GetContentModeration](#) AWS CLI

## get-face-detection

Il seguente esempio di codice mostra come utilizzare `get-face-detection`.

### AWS CLI

Per ottenere i risultati di un'operazione di riconoscimento facciale

Il `get-face-detection` comando seguente mostra i risultati di un'operazione di riconoscimento facciale avviata in precedenza chiamando `start-face-detection`.

```
aws rekognition get-face-detection \  
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Output:

```
{  
  "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":
"0zL223pDKy91160/02KXRqFIEAwxy4PkgYcm3hSo0rdysbXg5Ex0eFgTGEj0ADEac6S037U",
"JobStatus": "SUCCEEDED",
"VideoMetadata": {
    "Format": "QuickTime / MOV",
```

```
    "FrameRate": 29.970617294311523,  
    "Codec": "h264",  
    "DurationMillis": 6806,  
    "FrameHeight": 1080,  
    "FrameWidth": 1920  
  }  
}
```

Per ulteriori informazioni, [consulta Rilevamento di volti in un video archiviato](#) nella Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta [GetFaceDetection](#) Command Reference.AWS CLI

## get-face-search

Il seguente esempio di codice mostra come utilizzare `get-face-search`.

### AWS CLI

Per ottenere i risultati di un'operazione di ricerca facciale

Il `get-face-search` comando seguente mostra i risultati di un'operazione di ricerca facciale avviata in precedenza chiamando `start-face-search`.

```
aws rekognition get-face-search \  
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

Output:

```
{  
  "Persons": [  
    {  
      "Timestamp": 467,  
      "FaceMatches": [],  
      "Person": {  
        "Index": 0,  
        "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,
  "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
}
}
}

```

Per ulteriori informazioni, consulta la sezione [Ricerca di volti nei video archiviati](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [GetFaceSearch](#) Reference AWS CLI .

## get-label-detection

Il seguente esempio di codice mostra come utilizzare `get-label-detection`.

## AWS CLI

Per ottenere i risultati di un'operazione di rilevamento di oggetti e scene

Il `get-label-detection` comando seguente visualizza i risultati di un'operazione di rilevamento di oggetti e scene avviata in precedenza chiamando `start-label-detection`.

```
aws rekognition get-label-detection \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

Output:

```
{
  "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": "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": "BMugzAi4L72IERzQdbpyMQuEFBsjlo5W0Yx3mfG+sR9mm98E1/
Cp0benspRfs/5FBQFs4X7G"
}

```

Per ulteriori informazioni, consulta [Detecting Labels in a Video nella Amazon Rekognition Developer Guide](#).

- Per i API dettagli, consulta [GetLabelDetection](#) Command Reference.AWS CLI

## get-person-tracking

Il seguente esempio di codice mostra come utilizzare `get-person-tracking`.

### AWS CLI

Per ottenere i risultati di un'operazione di People Pathing

Il `get-person-tracking` comando seguente visualizza i risultati di un'operazione di tracciamento delle persone avviata in precedenza chiamando `start-person-tracking`

```
aws rekognition get-person-tracking \
  --job-id 1234567890abcdef1234567890abcdef1234567890abcdef
```

Output:

```
{
```

```
"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
}
}
```

Per ulteriori informazioni, consulta [People Pathing](#) nella Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta [GetPersonTracking](#) Command Reference.AWS CLI

## index-faces

Il seguente esempio di codice mostra come utilizzare `index-faces`.

Per ulteriori informazioni, consulta [Indicizzazione dei volti in una raccolta](#).

### AWS CLI

Per aggiungere volti a una raccolta

Il `index-faces` comando seguente aggiunge le facce trovate in un'immagine alla raccolta specificata.

```
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"
```

Output:

```
{  
  "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"
}
```

Per ulteriori informazioni, [consulta Adding Faces to a Collection](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [IndexFaces](#)Reference AWS CLI .

## list-collections

Il seguente esempio di codice mostra come utilizzare `list-collections`.

Per ulteriori informazioni, consulta [Creazione dell'elenco delle raccolte](#).

### AWS CLI

Per elencare le raccolte disponibili

Il `list-collections` comando seguente elenca le raccolte disponibili nell' AWS account.

```
aws rekognition list-collections
```

Output:

```
{
  "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"
  ]
}
```

Per ulteriori informazioni, consulta [Listing Collections](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [ListCollections](#)Reference AWS CLI .

## list-faces

Il seguente esempio di codice mostra come utilizzare `list-faces`.

Per ulteriori informazioni, consulta [Creazione dell'elenco dei volti in una raccolta](#).

### AWS CLI

Per elencare i volti di una raccolta

Il `list-faces` comando seguente elenca le facce della raccolta specificata.

```
aws rekognition list-faces \  
  --collection-id MyCollection
```

## Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, [consulta Listing Faces in a Collection](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [ListFaces](#)Reference AWS CLI .

## list-stream-processors

Il seguente esempio di codice mostra come utilizzare `list-stream-processors`.

### AWS CLI

Per elencare gli stream processor presenti nel tuo account

Il `list-stream-processors` comando seguente elenca gli stream processor presenti nel tuo account e lo stato di ciascuno.

```
aws rekognition list-stream-processors
```

Output:

```
{
  "StreamProcessors": [
    {
      "Status": "STOPPED",
      "Name": "my-stream-processor"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Lavorare con i video in streaming](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [ListStreamProcessors](#) Reference AWS CLI .

## recognize-celebrities

Il seguente esempio di codice mostra come utilizzare `recognize-celebrities`.

Per ulteriori informazioni, consulta [Riconoscimento delle celebrità in un'immagine](#).

### AWS CLI

Riconoscere le celebrità in un'immagine

Il `recognize-celebrities` comando seguente riconosce le celebrità nell'immagine specificata archiviata in un bucket Amazon S3. :

```
aws rekognition recognize-celebrities \  
--image "S3Object={Bucket=MyImageS3Bucket,Name=moviestars.jpg}"
```

Output:

```
{  
  "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"  
        }  
      ]  
    }  
  ]  
}
```



```
        "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Riconoscere le celebrità in un'immagine nella Amazon Rekognition Developer Guide](#).

- Per i API dettagli, consulta Command Reference. [RecognizeCelebritiesAWS CLI](#)

## search-faces-by-image

Il seguente esempio di codice mostra come utilizzare `search-faces-by-image`.

Per ulteriori informazioni, consulta [Ricerca di un volto \(immagine\)](#).

### AWS CLI

Per cercare volti in una raccolta che corrispondano al volto più grande di un'immagine.

Il `search-faces-by-image` comando seguente cerca i volti in una raccolta che corrispondono al volto più grande dell'immagine specificata. :

```

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"  
}
```

Per ulteriori informazioni, consulta [Searching for a face using an image](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [SearchFacesByImage](#)Reference AWS CLI .

## search-faces

Il seguente esempio di codice mostra come utilizzare `search-faces`.

Per ulteriori informazioni, consulta [Ricerca di un volto \(ID volto\)](#).

### AWS CLI

Per cercare volti in una raccolta che corrispondano a un Face ID.

Il `search-faces` comando seguente cerca i volti in una raccolta che corrispondono al face ID specificato.

```
aws rekognition search-faces \  
  --face-id 8d3cfc70-4ba8-4b36-9644-90fba29c2dac \  
  --collection-id MyCollection
```

Output:

```
{  
  "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
}
]
}

```

Per ulteriori informazioni, consulta la sezione [Ricerca di un volto utilizzando il relativo Face ID](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [SearchFaces](#)Reference AWS CLI .

## start-celebrity-recognition

Il seguente esempio di codice mostra come utilizzare `start-celebrity-recognition`.

### AWS CLI

Per avviare il riconoscimento delle celebrità in un video memorizzato

Il `start-celebrity-recognition` comando seguente avvia un processo per cercare celebrità nel file video specificato archiviato in un bucket Amazon S3.

```

aws rekognition start-celebrity-recognition \
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"

```

Output:

```
{
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Per ulteriori informazioni, consulta [Riconoscere le celebrità in un video archiviato nella](#) Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta Command Reference. [StartCelebrityRecognition](#)AWS CLI

## start-content-moderation

Il seguente esempio di codice mostra come utilizzare `start-content-moderation`.

AWS CLI

Per avviare il riconoscimento di contenuti non sicuri in un video archiviato

Il `start-content-moderation` comando seguente avvia un processo per rilevare contenuti non sicuri nel file video specificato archiviato in un bucket Amazon S3.

```
aws rekognition start-content-moderation \
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Output:

```
{
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Per ulteriori informazioni, consulta [Rilevamento di video archiviati non sicuri](#) nella Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta Command Reference. [StartContentModeration](#)AWS CLI

## start-face-detection

Il seguente esempio di codice mostra come utilizzare `start-face-detection`.

## AWS CLI

Per rilevare i volti in un video

Il `start-face-detection` comando seguente avvia un processo per rilevare i volti nel file video specificato archiviato in un bucket Amazon S3.

```
aws rekognition start-face-detection
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Output:

```
{
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Per ulteriori informazioni, [consulta Rilevamento di volti in un video archiviato](#) nella Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta [StartFaceDetection](#) Command Reference.AWS CLI

## `start-face-search`

Il seguente esempio di codice mostra come utilizzare `start-face-search`.

## AWS CLI

Per cercare volti in una raccolta che corrispondano ai volti rilevati in un video

Il `start-face-search` comando seguente avvia un processo per la ricerca di volti in una raccolta che corrispondono ai volti rilevati nel file video specificato in un bucket Amazon S3.

```
aws rekognition start-face-search \
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}" \
  --collection-id collection
```

Output:

```
{
```

```
"JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

Per ulteriori informazioni, consulta la sezione [Ricerca di volti nei video archiviati](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [StartFaceSearch](#)Reference AWS CLI .

## start-label-detection

Il seguente esempio di codice mostra come utilizzare `start-label-detection`.

### AWS CLI

Per rilevare oggetti e scene in un video

Il `start-label-detection` comando seguente avvia un processo per rilevare oggetti e scene nel file video specificato archiviato in un bucket Amazon S3.

```
aws rekognition start-label-detection \  
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Output:

```
{  
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"  
}
```

Per ulteriori informazioni, consulta [Detecting Labels in a Video nella](#) Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta [StartLabelDetection](#)Command Reference.AWS CLI

## start-person-tracking

Il seguente esempio di codice mostra come utilizzare `start-person-tracking`.

### AWS CLI

Per avviare il percorso delle persone in un video memorizzato



Il `start-person-tracking` comando seguente avvia un processo per tenere traccia dei percorsi seguiti dagli utenti nel file video specificato archiviato in un bucket Amazon S3. :

```
aws rekognition start-person-tracking \  
  --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

Output:

```
{  
  "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"  
}
```

Per ulteriori informazioni, consulta [People Pathing](#) nella Amazon Rekognition Developer Guide.

- Per i API dettagli, consulta [StartPersonTracking](#) Command Reference.AWS CLI

## start-stream-processor

Il seguente esempio di codice mostra come utilizzare `start-stream-processor`.

AWS CLI

Per avviare uno stream processor

Il `start-stream-processor` comando seguente avvia il processore di streaming video specificato.

```
aws rekognition start-stream-processor \  
  --name my-stream-processor
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lavorare con i video in streaming](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [StartStreamProcessor](#) Reference AWS CLI .

## stop-stream-processor

Il seguente esempio di codice mostra come utilizzare `stop-stream-processor`.

## AWS CLI

Per arrestare uno stream processor in esecuzione

Il `stop-stream-processor` comando seguente arresta il processore stream in esecuzione specificato.

```
aws rekognition stop-stream-processor \  
  --name my-stream-processor
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Lavorare con i video in streaming](#) nella Amazon Rekognition Developer Guide.

- Per API i dettagli, consulta Command [StopStreamProcessor](#)Reference AWS CLI .

## AWS RAM esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS RAM.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, dove puoi trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **accept-resource-share-invitation**

Il seguente esempio di codice mostra come utilizzare `accept-resource-share-invitation`.

AWS CLI

Accettare un invito alla condivisione di risorse

L'accept-resource-share-invitationesempio seguente accetta l'invito alla condivisione delle risorse specificato. I responsabili dell'account invitato possono iniziare immediatamente a utilizzare le risorse della condivisione.

```
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
```

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [AcceptResourceShareInvitation](#) in AWS CLI Command Reference.

## associate-resource-share-permission

Il seguente esempio di codice mostra come utilizzare associate-resource-share-permission.

AWS CLI

Per associare un'autorizzazione RAM gestita a una condivisione di risorse

L'associate-resource-share-permissionesempio seguente sostituisce l'autorizzazione gestita esistente per il tipo di risorsa pertinente con l'autorizzazione gestita specificata. L'accesso a tutte le risorse del tipo di risorsa pertinente è regolato dalla nuova autorizzazione.

```
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
```

Output:

```
{  
  "returnValue": true  
}
```

- Per API i dettagli, vedere [AssociateResourceSharePermission](#) in AWS CLI Command Reference.

## associate-resource-share

Il seguente esempio di codice mostra come utilizzare `associate-resource-share`.

### AWS CLI

Esempio 1: associare una risorsa a una condivisione di risorse

L'associazione di risorse seguente aggiunge una configurazione di licenza alla condivisione di risorse specificata.

```
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
```

Output:

```
{  
  "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
      }
    ]
  }

```

Esempio 2: associare un principale a una condivisione di risorse

L'associate-resource-shareesempio seguente concede l'accesso per la condivisione di risorse specificata a tutti gli account dell'unità organizzativa specificata.

```

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

```

Output:

```

{
  "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"
    }
  ]
}

```

- Per API i dettagli, vedere [AssociateResourceShare](#) in AWS CLI Command Reference.

## create-resource-share

Il seguente esempio di codice mostra come utilizzare create-resource-share.

### AWS CLI

Esempio 1: creare una condivisione di risorse

L'`create-resource-share` seguente crea una condivisione di risorse vuota con il nome specificato. È necessario aggiungere separatamente risorse, principali e autorizzazioni alla condivisione.

```
aws ram create-resource-share \  
  --name MyNewResourceShare
```

Output:

```
{  
  "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  
  }  
}
```

Esempio 2: creare una condivisione di risorse con AWS account come responsabili

L'`create-resource-share` seguente crea una condivisione di risorse e concede l'accesso all' AWS account specificato (222222222222). Se i responsabili specificati non fanno parte della stessa AWS organizzazione, gli inviti vengono inviati e devono essere accettati prima di concedere l'accesso.

```
aws ram create-resource-share \  
  --name MyNewResourceShare \  
  --principals 222222222222
```

Esempio 3: creare una condivisione di risorse limitata alla propria organizzazione AWS

L'`create-resource-share` seguente crea una condivisione di risorse limitata agli account dell' AWS organizzazione di cui l'account è membro e aggiunge l'unità organizzativa specificata come principale. Tutti gli account di quell'unità organizzativa possono utilizzare le risorse della condivisione di risorse.

```
aws ram create-resource-share \  
  --principals 222222222222
```

```
--name MyNewResourceShare \  
--no-allow-external-principals \  
--principals arn:aws:organizations::123456789012:ou/o-63bEXAMPLE/ou-46xi-  
rEXAMPLE
```

Output:

```
{  
  "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  
  }  
}
```

- Per API i dettagli, vedere [CreateResourceShare](#) in AWS CLI Command Reference.

## delete-resource-share

Il seguente esempio di codice mostra come utilizzare `delete-resource-share`.

AWS CLI

Per eliminare una condivisione di risorse

L'`delete-resource-share` esempio seguente elimina la condivisione di risorse specificata.

```
aws ram delete-resource-share \  
--resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE
```

L'output seguente indica il successo:

```
{  
  "returnValue": true  
}
```

- Per API i dettagli, vedere [DeleteResourceShare](#) in AWS CLI Command Reference.

## disassociate-resource-share-permission

Il seguente esempio di codice mostra come utilizzare `disassociate-resource-share-permission`.

### AWS CLI

Per rimuovere un'autorizzazione RAM gestita per un tipo di risorsa da una condivisione di risorse

L'`disassociate-resource-share-permission` esempio seguente rimuove l'autorizzazione RAM gestita per i database Glue dalla condivisione di risorse specificata.

```
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
```

Output:

```
{
  "returnValue": true
}
```

- Per API i dettagli, vedere [DisassociateResourceSharePermission](#) in AWS CLI Command Reference.

## disassociate-resource-share

Il seguente esempio di codice mostra come utilizzare `disassociate-resource-share`.

### AWS CLI

Per rimuovere una risorsa da una condivisione di risorse

L'`disassociate-resource-share` esempio seguente rimuove la risorsa specificata, in questo caso una VPC sottorete, dalla condivisione di risorse specificata. I principali con accesso alla condivisione di risorse non possono più eseguire operazioni su quella risorsa.



```
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
```

Output:

```
{  
  "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  
  ]  
}
```

- Per API i dettagli, vedere [DisassociateResourceShare](#) in AWS CLI Command Reference.

## enable-sharing-with-aws-organization

Il seguente esempio di codice mostra come utilizzare `enable-sharing-with-aws-organization`.

AWS CLI

Per abilitare la condivisione delle risorse tra AWS Organizzazioni

L'esempio seguente consente la condivisione delle risorse tra l'organizzazione e le unità organizzative.

```
aws ram enable-sharing-with-aws-organization
```

Il seguente output indica il successo.

```
{  
  "returnValue": true  
}
```

- Per API i dettagli, vedere [EnableSharingWithAwsOrganization](#) in AWS CLI Command Reference.

## get-permission

Il seguente esempio di codice mostra come utilizzare `get-permission`.

### AWS CLI

Per recuperare i dettagli di un'autorizzazione RAM gestita

L'`get-permission` esempio seguente visualizza i dettagli per la versione predefinita dell'autorizzazione RAM gestita specificata.

```
aws ram get-permission \  
  --permission-arn arn:aws:ram::aws:permission/  
AWSRAMPermissionGlueTableReadWriteForDatabase
```

Output:

```
{  
  "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  
  }  
}
```

- Per API i dettagli, vedere [GetPermission](#) in AWS CLI Command Reference.

## get-resource-policies

Il seguente esempio di codice mostra come utilizzare `get-resource-policies`.

### AWS CLI

Per ottenere le politiche per una risorsa

L'esempio seguente visualizza le politiche di autorizzazione basate sulle risorse per la risorsa specificata associata a una condivisione di risorse.

```
aws ram get-resource-policies \
  --resource-arns arn:aws:ec2:us-west-2:123456789012:subnet/
  subnet-0250c25a1fEXAMPLE
```

Output:

```
{
  "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"
          ]
        }
      ]
    }
  ]
}
```

- Per API i dettagli, vedere [GetResourcePolicies](#) in AWS CLI Command Reference.

## get-resource-share-associations

Il seguente esempio di codice mostra come utilizzare `get-resource-share-associations`.

### AWS CLI

Esempio 1: Per elencare tutte le associazioni di risorse per tutti i tipi di risorse

L'esempio seguente elenca le associazioni di risorse per tutti i tipi di risorse in tutte le condivisioni di risorse.

```
aws ram get-resource-share-associations \
```

```
--association-type RESOURCE
```

Output:

```
{
  "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
    }
  ]
}
```

Esempio 2: Elencare le principali associazioni per una condivisione di risorse

L'get-resource-share-associationsesempio seguente elenca solo le associazioni principali solo per la condivisione di risorse specificata.

```
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
```

**Output:**

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [GetResourceShareAssociations](#) in AWS CLI Command Reference.

**get-resource-share-invitations**

Il seguente esempio di codice mostra come utilizzare `get-resource-share-invitations`.

**AWS CLI**

Per elencare gli inviti alla condivisione di risorse

L'`get-resource-share-invitations` esempio seguente elenca gli attuali inviti alla condivisione di risorse.

```
aws ram get-resource-share-invitations
```

**Output:**

```
{
  "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"
    }
]
}

```

- Per API i dettagli, vedere [GetResourceShareInvitations](#) in AWS CLI Command Reference.

## get-resource-shares

Il seguente esempio di codice mostra come utilizzare `get-resource-shares`.

### AWS CLI

Esempio 1: elencare le condivisioni di risorse che possiedi e condividi con altri

L'`get-resource-shares` esempio seguente elenca le condivisioni di risorse che sono state create e che vengono condivise con altri.

```

aws ram get-resource-shares \
  --resource-owner SELF

```

Output:

```

{
  "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
}
]
}

```

Esempio 2: Per elencare le condivisioni di risorse possedute da altri e condivise con te

L'get-resource-sharesesempio seguente elenca le condivisioni di risorse che altri hanno creato e condiviso con te. In questo esempio, non ce ne sono.

```

aws ram get-resource-shares \
  --resource-owner OTHER-ACCOUNTS

```

Output:

```

{
  "resourceShares": []
}

```

- Per API i dettagli, vedere [GetResourceShares](#) in AWS CLI Command Reference.

## list-pending-invitation-resources

Il seguente esempio di codice mostra come utilizzarelist-pending-invitation-resources.

AWS CLI

Per elencare le risorse disponibili in una condivisione di risorse in sospeso

L'`list-pending-invitation-resources` seguente elenca tutte le risorse presenti nella condivisione di risorse associata all'invito specificato.

```
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
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ListPendingInvitationResources](#) in AWS CLI Command Reference.

## list-permissions

Il seguente esempio di codice mostra come utilizzare `list-permissions`.



## AWS CLI

Per elencare le autorizzazioni RAM gestite disponibili

L'`list-permissions` seguente elenca tutte le autorizzazioni RAM gestite disponibili solo per il tipo di risorsa del database AWS Glue.

```
aws ram list-permissions \  
  --resource-type glue:Database
```

Output:

```
{  
  "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
    }
]
}

```

L'`list-permissionesempio` seguente mostra le autorizzazioni RAM gestite disponibili per tutti i tipi di risorse.

```
aws ram list-permissions
```

Output:

```

{
  "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
    }
]
}

```

- Per API i dettagli, vedere [ListPermissions](#) in AWS CLI Command Reference.

## list-principals

Il seguente esempio di codice mostra come utilizzare `list-principals`.

### AWS CLI

Elencare i principali con accesso a una risorsa

L'`list-principals` esempio seguente visualizza un elenco dei principali che possono accedere alle risorse del tipo specificato tramite qualsiasi condivisione di risorse.

```
aws ram list-principals \
  --resource-type ec2:Subnet
```

Output:

```
{
  "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
    }
  ]
}

```

- Per API i dettagli, vedere [ListPrincipals](#) in AWS CLI Command Reference.

## list-resource-share-permissions

Il seguente esempio di codice mostra come utilizzare `list-resource-share-permissions`.

### AWS CLI

Per elencare tutte le autorizzazioni RAM gestite attualmente associate a una condivisione di risorse

L'`list-resource-share-permissions` esempio seguente elenca tutte le autorizzazioni RAM gestite allegate alla condivisione di risorse specificata.

```

aws ram list-resource-share-permissions \
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE

```

Output:

```

{
  "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
}
]
```

- Per API i dettagli, vedere [ListResourceSharePermissions](#) in AWS CLI Command Reference.

## list-resource-types

Il seguente esempio di codice mostra come utilizzare `list-resource-types`.

### AWS CLI

Per elencare i tipi di risorse supportati da AWS RAM

L'`list-resource-types` esempio seguente elenca tutti i tipi di risorse attualmente supportati da AWS RAM.

```
aws ram list-resource-types
```

Output:

```
{
  "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"
    }
  ]
}

```

- Per API i dettagli, vedere [ListResourceTypes](#) in AWS CLI Command Reference.

## list-resources

Il seguente esempio di codice mostra come utilizzare `list-resources`.

### AWS CLI

Per elencare le risorse associate a una condivisione di risorse

L'`list-resources` esempio seguente elenca tutte le risorse della condivisione di risorse specificata che sono del tipo di risorsa specificato.

```

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

```

Output:

```

{
  "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
    }
  ]
}

```

- Per API i dettagli, vedere [ListResources](#) in AWS CLI Command Reference.

## **promote-resource-share-created-from-policy**

Il seguente esempio di codice mostra come utilizzare `promote-resource-share-created-from-policy`.

### AWS CLI

Promuovere una condivisione delle risorse basata sulla politica delle risorse alla piena funzionalità in AWS RAM

L'`promote-resource-share-created-from-policy` esempio seguente prende una condivisione di risorse creata implicitamente allegando una policy basata sulle risorse e la converte in modo che sia perfettamente funzionante con la console e le relative operazioni. 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
```

Output:

```
{  
  "returnValue": true  
}
```

- Per i API dettagli, vedere in Command Reference.  
[PromoteResourceShareCreatedFromPolicy](#) AWS CLI

## **reject-resource-share-invitation**

Il seguente esempio di codice mostra come utilizzare `reject-resource-share-invitation`.

### AWS CLI

Per rifiutare un invito alla condivisione di risorse

L'`reject-resource-share-invitation` esempio seguente rifiuta l'invito alla condivisione delle risorse specificato.

```
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
```

Output:

```
"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"  
  }  
]
```

- Per API i dettagli, vedere [RejectResourceShareInvitation](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere tag a una condivisione di risorse

L' `tag-resource` esempio seguente aggiunge una chiave di tag `project` e un valore associato `lima` alla condivisione di risorse specificata.

```
aws ram tag-resource \  
  --tags key=project,value=lima \  
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.



## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una condivisione di risorse

L'`untag-resource` esempio seguente rimuove la chiave del `project` tag e il valore associato dalla condivisione di risorse specificata.

```
aws ram untag-resource \  
  --tag-keys project \  
  --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-resource-share

Il seguente esempio di codice mostra come utilizzare `update-resource-share`.

### AWS CLI

Per aggiornare una condivisione di risorse

L'`update-resource-share` esempio seguente modifica la condivisione di risorse specificata per consentire i principali esterni che non fanno parte di un' AWS organizzazione.

```
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
```

Output:

```
{  
  "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
  }
}
```

- Per API i dettagli, vedere [UpdateResourceShare](#) in AWS CLI Command Reference.

## Esempi di Resource Explorer utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Resource Explorer.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni illustrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto degli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **associate-default-view**

Il seguente esempio di codice mostra come utilizzare `associate-default-view`.

AWS CLI

Per impostare una visualizzazione di Resource Explorer come predefinita per la relativa AWS regione

L'`associate-default-view` esempio seguente imposta una vista, come specificato da `itsARN`, come visualizzazione predefinita per la AWS regione in cui viene chiamata l'operazione.

```
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
```

Output:

```
{  
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-  
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"  
}
```

Per ulteriori informazioni, vedere [Impostazione di una visualizzazione predefinita in una AWS regione nella Guida per l'utente di AWS Resource Explorer](#).

- Per API i dettagli, vedere [AssociateDefaultView](#) in AWS CLI Command Reference.

## batch-get-view

Il seguente esempio di codice mostra come utilizzare `batch-get-view`.

### AWS CLI

Per recuperare dettagli su più visualizzazioni di Resource Explorer

L'esempio seguente mostra i dettagli relativi a due viste specificate da loro ARNs. Utilizzate gli spazi per separare i multipli ARNs nel parametro `--view-arn`.

```
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
```

Output:

```
{  
  "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": []
}

```

Per ulteriori informazioni sulle visualizzazioni, vedere [Informazioni sulle visualizzazioni di Resource Explorer nella Guida per gli utenti di AWS Resource Explorer](#).

- Per API i dettagli, vedere [BatchGetView](#) in AWS CLI Command Reference.

## create-index

Il seguente esempio di codice mostra come utilizzare `create-index`.

### AWS CLI

Per attivare Resource Explorer in una AWS regione creando un indice

L'`create-index`esempio seguente crea un indice locale nella AWS regione in cui viene chiamata l'operazione. Genera AWS CLI automaticamente un valore di `client-token` parametro casuale e lo include nella chiamata a AWS se non si specifica un valore.

```
aws resource-explorer-2 create-index \  
  --region us-east-1
```

Output:

```
{  
  "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"  
}
```

Dopo aver creato un indice locale, puoi convertirlo nell'indice di aggregazione per l'account eseguendo il [update-index-type](#) comando.

Per ulteriori informazioni, consulta [Attivazione di Resource Explorer in una AWS regione per indicizzare le risorse](#) nella Guida per l'utente di AWS Resource Explorer.

- Per API i dettagli, consulta [CreateIndex AWS CLI](#) Command Reference.

## create-view

Il seguente esempio di codice mostra come utilizzare `create-view`.

### AWS CLI

Esempio 1: creare una vista non filtrata per l'indice in una regione AWS

L'`create-view`esempio seguente crea una vista nella AWS regione specificata che restituisce tutti i risultati nella regione senza alcun filtro. La visualizzazione include il campo opzionale `Tag` sui risultati restituiti. Poiché questa vista viene creata nella regione che contiene l'indice di aggregazione, può includere i risultati di tutte le regioni dell'account che contengono un indice Resource Explorer.

```
aws resource-explorer-2 create-view \  
  --view-name My-Main-View \  
  --included-properties Name=tags \  
  --region us-east-1
```

```
--region us-east-1
```

Output:

```
{
  "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"
  }
}
```

Esempio 2: creare una visualizzazione che restituisca solo le risorse associate ad Amazon EC2

Quanto segue `create-view` crea una visualizzazione nella AWS regione `us-east-1` che restituisce solo le risorse della regione associate al EC2 servizio Amazon. La visualizzazione include il `Tags` campo opzionale sui risultati restituiti. Poiché questa vista viene creata nella regione che contiene l'indice dell'aggregatore, può includere i risultati di tutte le regioni dell'account che contengono un indice Resource Explorer.

```
aws resource-explorer-2 create-view \
  --view-name My-EC2-Only-View \
  --included-properties Name=tags \
  --filters FilterString="service:ec2" \
  --region us-east-1
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Creazione di viste per la ricerca nella Guida per l'utente di AWS Resource Explorer](#).

- Per API i dettagli, vedere [CreateView](#) in AWS CLI Command Reference.

## delete-index

Il seguente esempio di codice mostra come utilizzare `delete-index`.

### AWS CLI

Per disattivare Resource Explorer in una AWS regione eliminandone l'indice

L'`delete-index` esempio seguente elimina l'indice Resource Explorer specificato nella AWS regione in cui viene effettuata la richiesta.

```

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

```

Output:

```

{
  "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-
cdef-fedc-EXAMPLE22222",
  "State": "DELETING"
}

```

Per ulteriori informazioni sull'eliminazione di un indice, vedere [Disattivazione di AWS Resource Explorer in una AWS regione nella Guida](#) per l'utente di AWS Resource Explorer.

- Per API i dettagli, vedere [DeleteIndex](#) in AWS CLI Command Reference.

## **delete-view**

Il seguente esempio di codice mostra come utilizzare `delete-view`.

### AWS CLI

Per eliminare una visualizzazione di Resource Explorer

L'`delete-view` esempio seguente elimina una visualizzazione specificata da itsARN.

```
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
```

Output:

```
{  
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"  
}
```

Per ulteriori informazioni, vedere [Eliminazione delle viste](#) nella Guida per l'utente di AWS Resource Explorer.

- Per API i dettagli, vedere [DeleteView](#) in AWS CLI Command Reference.

## **disassociate-default-view**

Il seguente esempio di codice mostra come utilizzare `disassociate-default-view`.

### AWS CLI

Per rimuovere la visualizzazione predefinita di Resource Explorer per una AWS regione

Di seguito viene `disassociate-default-view` rimossa la visualizzazione predefinita di Resource Explorer per la AWS regione in cui viene richiamata l'operazione. Dopo aver eseguito



questa operazione, tutte le operazioni di ricerca nella regione devono specificare in modo esplicito una vista o l'operazione ha esito negativo.

```
aws resource-explorer-2 disassociate-default-view
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Impostazione di una visualizzazione predefinita in una AWS regione nella Guida per l'utente di AWS Resource Explorer](#).

- Per API i dettagli, vedere [DisassociateDefaultView](#) in AWS CLI Command Reference.

## get-default-view

Il seguente esempio di codice mostra come utilizzare `get-default-view`.

### AWS CLI

Per recuperare la vista Resource Explorer, che è la vista predefinita per la relativa regione AWS

L'`get-default-view` esempio seguente recupera ARN la visualizzazione predefinita per la AWS regione in cui viene chiamata l'operazione.

```
aws resource-explorer-2 get-default-view
```

Output:

```
{
  "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/default-view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

Per ulteriori informazioni, vedere [Impostazione di una visualizzazione predefinita in una AWS regione nella Guida per l'utente di AWS Resource Explorer](#).

- Per API i dettagli, vedere [GetDefaultView](#) in AWS CLI Command Reference.

## get-index

Il seguente esempio di codice mostra come utilizzare `get-index`.

## AWS CLI

Esempio 1: per recuperare i dettagli per un indice di aggregazione di Resource Explorer

L'`get-index` seguente visualizza i dettagli dell'indice Resource Explorer nella regione specificata AWS . Poiché la regione specificata contiene l'indice di aggregazione per l'account, l'output elenca le regioni che replicano i dati nell'indice di questa regione.

```
aws resource-explorer-2 get-index \  
  --region us-east-1
```

Output:

```
{  
  "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"  
}
```

Esempio 2: per recuperare i dettagli per un indice locale di Resource Explorer

L'`get-index` seguente visualizza i dettagli dell'indice Resource Explorer nella AWS regione specificata. Poiché la regione specificata contiene un indice locale, l'output elenca la regione in cui replica i dati dall'indice di questa regione.

```
aws resource-explorer-2 get-index \  
  --region us-west-2
```

Output:

```
{  
  "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"
  }

```

Per ulteriori informazioni sugli indici, consulta [Verifica delle AWS regioni in cui è attivo Resource Explorer](#) nella Guida per l'utente di AWS Resource Explorer.

- Per API i dettagli, vedere [GetIndex](#) in AWS CLI Command Reference.

## get-view

Il seguente esempio di codice mostra come utilizzare `get-view`.

### AWS CLI

Per recuperare i dettagli su una visualizzazione di Resource Explorer

L'`get-view` esempio seguente mostra i dettagli su una vista specificata da `itsARN`.

```

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

```

Output:

```

{
  "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"
}
}
```

Per ulteriori informazioni sulle visualizzazioni, vedere [Informazioni sulle visualizzazioni di Resource Explorer](#) nella Guida per l'utente di AWS Resource Explorer.

- Per API i dettagli, vedere [GetView](#) in AWS CLI Command Reference.

## list-indexes

Il seguente esempio di codice mostra come utilizzare `list-indexes`.

### AWS CLI

Per elencare le AWS regioni in cui Resource Explorer ha indici

L'`list-indexes` seguente elenca gli indici di tutte le regioni in cui Resource Explorer ha un indice. La risposta specifica il tipo di ciascun indice, la relativa AWS regione e la relativa ARN

```
aws resource-explorer-2 list-indexes
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni sugli indici, consulta [Verifica delle AWS aree in cui è attivo Resource Explorer](#) nella Guida per l'utente di AWS Resource Explorer.

- Per API i dettagli, vedere [ListIndexes](#) in AWS CLI Command Reference.

## list-supported-resource-types

Il seguente esempio di codice mostra come utilizzare `list-supported-resource-types`.

### AWS CLI

Per elencare le AWS regioni in cui Resource Explorer ha indici

L'`list-supported-resource-types` esempio seguente elenca tutti i tipi di risorse attualmente supportati da `AREXLong`. La risposta di esempio include un `NextToken` valore che indica che c'è più output disponibile da recuperare con chiamate aggiuntive.

```

aws resource-explorer-2 list-supported-resource-types \
  --max-items 10

```

Output:

```

{
  "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="
}

```

Per ottenere la parte successiva dell'output, richiamate nuovamente l'operazione e passate il valore di NextToken risposta della chiamata precedente come valore per `--starting-token`. Ripetere l'operazione finché NextToken è assente dalla risposta.

```

aws resource-explorer-2 list-supported-resource-types \
  --max-items 10 \

```

**--starting-****token** `eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0=`

Output:

```
{
  "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="
}
```

Per ulteriori informazioni sugli indici, consulta [Verifica delle AWS aree in cui è attivo Resource Explorer](#) nella Guida per l'utente di AWS Resource Explorer.

- Per API i dettagli, vedere [ListSupportedResourceTypes](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag associati a una visualizzazione o a un indice di Resource Explorer

L'`list-tags-for-resource` esempio seguente elenca le coppie di tag chiave e valore allegate alla visualizzazione con il valore specificato ARN. È necessario chiamare l'operazione dalla AWS regione che contiene la risorsa.

```
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
```

Output:

```
{
  "Tags": {
    "application": "MainCorpApp",
    "department": "1234"
  }
}
```

Per ulteriori informazioni sull'assegnazione di tag alle viste, consulta [Tagging views for access control](#) nella AWS Resource Explorer Users Guide.

- Per API i dettagli, consulta AWS CLI Command [ListTagsForResource](#) Reference.



## list-views

Il seguente esempio di codice mostra come utilizzare `list-views`.

### AWS CLI

Per elencare le viste di Resource Explorer disponibili in una AWS regione

L'`list-views` esempio seguente elenca tutte le viste disponibili nella regione in cui si richiama l'operazione.

```
aws resource-explorer-2 list-views
```

Output:

```
{
  "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"
  ]
}
```

Per ulteriori informazioni sulle visualizzazioni, vedere [Informazioni sulle visualizzazioni di Resource Explorer](#) nella Guida per l'utente di AWS Resource Explorer.

- Per API i dettagli, vedere [ListViews](#) in AWS CLI Command Reference.

## search

Il seguente esempio di codice mostra come utilizzare `search`.

### AWS CLI

Esempio 1: Per effettuare una ricerca utilizzando la visualizzazione predefinita

L'`search` esempio seguente visualizza tutte le risorse dell'area specificata associate al servizio. La ricerca utilizza la visualizzazione predefinita per la regione. La risposta di esempio include un `NextToken` valore che indica che è disponibile più output da recuperare con chiamate aggiuntive.

```
aws resource-explorer-2 search \  
--query-string "service:iam"
```

Output:

```
{  
  "Count": {  
    "Complete": true,  
    "TotalResources": 55  
  },  
  "NextToken":  
  "AG9V0EF1KLEXAMPLE0hJHVwo5chEXAMPLER5XiEpNrIgsEXAMPLE...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"  
}
```

Esempio 2: Effettuare una ricerca utilizzando una visualizzazione specificata

La ricerca di `search` esempio seguente mostra tutte le risorse («\*») nella AWS regione specificata che sono visibili attraverso la vista specificata. I risultati includono solo le risorse associate ad Amazon a EC2 causa dei filtri allegati alla vista.

```
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
```

## Output:

```
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": "2021-11-15T15:11:05Z",
    "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"
}

```

Per ulteriori informazioni, consulta [Using AWS Resource Explorer per la ricerca di AWS risorse](#) nella Resource Explorer Users Guide.

- Per API i dettagli, consulta [Search](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

## AWS CLI

Per etichettare una vista di Resource Explorer

L'`tag-resource` seguente aggiunge la chiave di tag «environment» con il valore «production» alla vista con il valore specificato ARN.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare le viste per il controllo degli accessi](#) nella Guida per l'utente di AWS Resource Explorer.

- Per API i dettagli, consulta [TagResource AWS CLI](#) Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

## AWS CLI

Per rimuovere un tag da una visualizzazione di Resource Explorer

L'`untag-resource` seguente rimuove qualsiasi tag con il nome chiave «environment» dalla vista con il nome specificato ARN.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare le viste per il controllo degli accessi](#) nella Guida per l'utente di AWS Resource Explorer.

- Per API i dettagli, consulta [UntagResource AWS CLI](#) Command Reference.

## update-index-type

Il seguente esempio di codice mostra come utilizzare `update-index-type`.

### AWS CLI

Per modificare il tipo di un indice di Resource Explorer

L'esempio seguente converte l'indice specificato da tipo `local aggregator` a tipo per attivare la possibilità di cercare risorse in tutte le AWS regioni dell'account. È necessario inviare la richiesta alla AWS regione che contiene l'indice che si desidera aggiornare.

```
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
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni sulla modifica del tipo di indice, consulta [Attivazione della ricerca tra aree geografiche creando un indice di aggregazione](#) nella Guida per gli utenti di AWS Resource Explorer.

- Per API i dettagli, consulta AWS CLI Command [UpdateIndexType](#) Reference.

## update-view

Il seguente esempio di codice mostra come utilizzare `update-view`.

## AWS CLI

Esempio 1: per aggiornare il `IncludedProperties` campo per una visualizzazione di Resource Explorer

L'`update-views` comando seguente aggiorna la vista specificata `tags` aggiungendola a quella opzionale `IncludedProperties`. Dopo aver eseguito questa operazione, le operazioni di ricerca che utilizzano questa visualizzazione includono informazioni sui tag associati alle risorse che appaiono nei risultati.

```
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
```

Output:

```
{  
  "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"  
  }  
}
```

Esempio 2: per aggiornare i filtri allegati a una vista

L'`update-views` comando seguente aggiorna la vista specificata per utilizzare un filtro che limita i risultati solo ai tipi di risorse associati al EC2 servizio Amazon.

```
aws resource-explorer-2 update-view \  
  --filter-string "type=EC2" \  
  --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/  
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222
```

```
--filters FilterString="service:ec2" \  
--view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View/  
EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni sulle visualizzazioni, consulta [Informazioni sulle visualizzazioni di Resource Explorer](#) nella Guida per gli utenti di AWS Resource Explorer.

- Per API i dettagli, vedere [UpdateView](#) in AWS CLI Command Reference.

## Esempi di Resource Groups utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con Resource Groups.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni illustrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)



## Azioni

### create-group

Il seguente esempio di codice mostra come utilizzare `create-group`.

#### AWS CLI

Esempio 1: creare un gruppo di risorse basato su tag

L'esempio seguente crea un gruppo di risorse basato su tag di EC2 istanze Amazon nella regione corrente. Si basa su una query relativa alle risorse etichettate con la chiave `Name` il valore `WebServers`. Il nome del gruppo è `tbq-WebServer`. La query si trova in un JSON file separato che viene passato al comando.

```
aws resource-groups create-group \  
  --name tbq-WebServer \  
  --resource-query file://query.json
```

Contenuto di `query.json`.

```
{  
  "Type": "TAG_FILTERS_1_0",  
  "Query": "{\"ResourceTypeFilters\": [\"AWS::EC2::Instance\"], \"TagFilters\":  
  [{\"Key\": \"Name\", \"Values\": [\"WebServers\"]}]}"  
}
```

Output:

```
{  
  "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\"]}]}"  
  }  
}
```

## Esempio 2: creare un gruppo di risorse CloudFormation basato sullo stack

L'`create-group` seguente crea un gruppo di risorse AWS CloudFormation basato sullo stack denominato `sampleCFNstackgroup`. La query include tutte le risorse nello CloudFormation stack specificato supportate da AWS Resource Groups.

```
aws resource-groups create-group \  
  --name cbq-CFNstackgroup \  
  --resource-query file://query.json
```

Contenuto di `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\"}"  
}
```

Output:

```
{  
  "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\"}"  
  }  
}
```

Per ulteriori informazioni, consulta [Create Groups](#) nella AWS Resource Groups User Guide.

- Per API i dettagli, vedere [CreateGroup](#) in AWS CLI Command Reference.

## `delete-group`

Il seguente esempio di codice mostra come utilizzare `delete-group`.

## AWS CLI

Per aggiornare la descrizione di un gruppo di risorse

L'`delete-group`esempio seguente aggiorna il gruppo di risorse specificato.

```
aws resource-groups delete-group \  
  --group-name tbq-WebServer
```

Output:

```
{  
  "Group": {  
    "GroupArn": "arn:aws:resource-groups:us-west-2:1234567890:group/tbq-  
WebServer",  
    "Name": "tbq-WebServer"  
  }  
}
```

Per ulteriori informazioni, consulta [Delete Groups](#) nella AWS Resource Groups User Guide.

- Per API i dettagli, vedere [DeleteGroup](#) in AWS CLI Command Reference.

## `get-group-query`

Il seguente esempio di codice mostra come utilizzare `get-group-query`.

## AWS CLI

Per allegare la query a un gruppo di risorse

L'`get-group-query`esempio seguente visualizza una query associata al gruppo di risorse specificato.

```
aws resource-groups get-group-query \  
  --group-name tbq-WebServer
```

Output:

```
{  
  "GroupQuery": {
```

```

    "GroupName": "tbq-WebServer",
    "ResourceQuery": {
      "Type": "TAG_FILTERS_1_0",
      "Query": "{\"ResourceTypeFilters\": [\"AWS::EC2::Instance\"], \"TagFilters\": [{\"Key\": \"Name\", \"Values\": [\"WebServers\"]}]}"
    }
  }
}

```

- Per API i dettagli, vedere [GetGroupQuery](#) in AWS CLI Command Reference.

## get-group

Il seguente esempio di codice mostra come utilizzare `get-group`.

### AWS CLI

Per ottenere informazioni su un gruppo di risorse

L'`get-group` esempio seguente mostra i dettagli sul gruppo di risorse specificato. Per allegare la query al gruppo, utilizzare `get-group-query`.

```

aws resource-groups get-group \
  --group-name tbq-WebServer

```

Output:

```

{
  "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."
  }
}

```

- Per API i dettagli, vedere [GetGroup](#) in AWS CLI Command Reference.

## get-tags

Il seguente esempio di codice mostra come utilizzare `get-tags`.

## AWS CLI

Per recuperare i tag allegati a un gruppo di risorse

L'get-tags esempio seguente visualizza le coppie di tag chiave e valore associate al gruppo di risorse specificato (il gruppo stesso, non i suoi membri).

```
aws resource-groups get-tags \  
--arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer
```

Output:

```
{  
  "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",  
  "Tags": {  
    "QueryType": "tags",  
    "QueryResources": "ec2-instances"  
  }  
}
```

- Per API i dettagli, vedere [GetTags](#) in AWS CLI Command Reference.

## list-group-resources

Il seguente esempio di codice mostra come utilizzare list-group-resources.

## AWS CLI

Per elencare tutte le risorse di un gruppo di risorse

Esempio 1: L'list-resource-group esempio seguente elenca tutte le risorse che fanno parte del gruppo di risorse specificato.

```
aws resource-groups list-group-resources \  
--group-name tbq-WebServer
```

Output:

```
{  
  "ResourceIdentifiers": [  
    {
```

```

        "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-09f77fa38c12345ab",
        "ResourceType": "AWS::EC2::Instance"
    }
]
}

```

Esempio 2: L'esempio seguente elenca tutte le risorse del gruppo che hanno anche un «tipo di risorsa» del tipo '::: :Instance'AWS EC2 . :

```
aws resource-groups list-group-resources --group-name tbq- WebServer --filters Nome=Tipo di
risorsa, Valori=::: :Instance AWS EC2
```

- Per API i dettagli, consulta [ListGroupResources AWS CLI](#) Command Reference.

## list-groups

Il seguente esempio di codice mostra come utilizzare `list-groups`.

### AWS CLI

Per elencare i gruppi di risorse disponibili

L'`list-groups`esempio seguente visualizza un elenco di tutti i gruppi di risorse.

```
aws resource-groups list-groups
```

Output:

```

{
  "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"
  }
]
```

- Per API i dettagli, vedere [ListGroups](#) in AWS CLI Command Reference.

## list-resource-groups

Il seguente esempio di codice mostra come utilizzare `list-resource-groups`.

### AWS CLI

Per elencare tutte le risorse di un gruppo di risorse

L'`list-resource-groups` esempio seguente elenca tutte le risorse che fanno parte del gruppo di risorse specificato.

```
aws resource-groups list-group-resources \
  --group-name tbq-WebServer
```

Output:

```
{
  "ResourceIdentifiers": [
    {
      "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/i-09f77fa38c12345ab",
      "ResourceType": "AWS::EC2::Instance"
    }
  ]
}
```

- Per API i dettagli, vedere [ListResourceGroups](#) in AWS CLI Command Reference.

## put-group-configuration

Il seguente esempio di codice mostra come utilizzare `put-group-configuration`.

### AWS CLI

Per allegare una configurazione di servizio a un gruppo di risorse

Esempio 1: l'`put-group-configuration` seguente specifica che il gruppo di risorse deve contenere solo le prenotazioni di EC2 capacità Amazon per le istanze delle famiglie C5 or M5.

```
aws resource-groups put-group-configuration \  
  --group MyTestGroup \  
  --configuration file://config.json
```

Contenuto di `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" ]
  }
]

```

Questo comando non produce alcun output in caso di successo.

Per ulteriori informazioni, vedere [Configurazioni dei servizi per i gruppi di risorse](#) nella Resource Groups API Reference Guide.

- Per API i dettagli, vedere [PutGroupConfiguration](#) in AWS CLI Command Reference.

## search-resources

Il seguente esempio di codice mostra come utilizzare `search-resources`.

### AWS CLI

Per trovare risorse che corrispondono a una query

L'esempio seguente recupera un elenco di tutte le AWS risorse che corrispondono alla query specificata.

```

aws resource-groups search-resources \
  --resource-query file://query.json

```

Contenuto di `query.json`.

```

{
  "Type": "TAG_FILTERS_1_0",
  "Query": "{\"ResourceTypeFilters\": [\"AWS::EC2::Instance\"], \"TagFilters\": [ { \"Key\": \"Patch Group\", \"Values\": [ \"Dev\" ] } ]}"
}

```

Output:

```

{
  "ResourceIdentifiers": [
    {

```

```

    "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-01a23bc45d67890ef",
    "ResourceType": "AWS::EC2::Instance"
  }
]
}

```

- Per API i dettagli, vedere [SearchResources](#) in AWS CLI Command Reference.

## tag

Il seguente esempio di codice mostra come utilizzare `tag`.

### AWS CLI

Per allegare un tag a un gruppo di risorse

L'etichetta seguente associa le coppie di chiave e valore del tag specificate al gruppo di risorse specificato (il gruppo stesso, non i suoi membri).

```

aws resource-groups tag \
  --tags QueryType=tags, QueryResources=ec2-instances \
  --arn arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer

```

Output:

```

{
  "Arn": "arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer",
  "Tags": {
    "QueryType": "tags",
    "QueryResources": "ec2-instances"
  }
}

```

Per ulteriori informazioni, consulta [Manage tags](#) nella AWS Resource Groups User Guide.

- Per API i dettagli, consulta [Tag](#) in AWS CLI Command Reference.

## untag

Il seguente esempio di codice mostra come utilizzare `untag`.

## AWS CLI

Per rimuovere i tag da un gruppo di risorse

L'untagsesempio seguente rimuove qualsiasi tag con la chiave specificata dal gruppo di risorse stesso, non dai suoi membri.

```
aws resource-groups untag \  
  --arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer \  
  --keys QueryType
```

Output:

```
{  
  "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",  
  "Keys": [  
    "QueryType"  
  ]  
}
```

Per ulteriori informazioni, consulta [Manage tags](#) nella AWS Resource Groups User Guide.

- Per API i dettagli, consulta [Untag](#) in AWS CLI Command Reference.

## update-group-query

Il seguente esempio di codice mostra come usare `update-group-query`.

### AWS CLI

Esempio 1: per aggiornare la query per un gruppo di risorse basato su tag

L'update-group-queryesempio seguente aggiorna la query allegata al gruppo di risorse basato su tag specificato.

```
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  
\"]}]}"'
```

Output:

```
{
  "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\"]}]}"
  }
}
```

Per ulteriori informazioni, consulta [Update Groups](#) nella AWS Resource Groups User Guide.

Esempio 2: per aggiornare la query per un gruppo di risorse CloudFormation basato sullo stack

L'update-group-queryesempio seguente aggiorna la query allegata al gruppo di risorse basato sullo AWS CloudFormation stack specificato.

```
aws resource-groups update-group-query \
  --group-name cbq-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\"}'
```

Output:

```
{
  "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\"}"
  }
}
```

Per ulteriori informazioni, consulta [Update Groups](#) nella AWS Resource Groups User Guide.

- Per API i dettagli, vedere [UpdateGroupQuery](#) in AWS CLI Command Reference.

## update-group

Il seguente esempio di codice mostra come utilizzare `update-group`.

### AWS CLI

Per aggiornare la descrizione di un gruppo di risorse

L'`update-group`esempio seguente aggiorna la descrizione per il gruppo di risorse specificato.

```
aws resource-groups update-group \  
  --group-name tbq-WebServer \  
  --description "Resource group for all web server resources."
```

Output:

```
{  
  "Group": {  
    "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-  
WebServer",  
    "Name": "tbq-WebServer"  
    "Description": "Resource group for all web server resources."  
  }  
}
```

Per ulteriori informazioni, consulta [Update Groups](#) nella AWS Resource Groups User Guide.

- Per API i dettagli, vedere [UpdateGroup](#) in AWS CLI Command Reference.

## API Esempi di etichettatura dei Resource Groups utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il tag AWS Command Line Interface API con Resource Groups.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **get-resources**

Il seguente esempio di codice mostra come utilizzare `get-resources`.

#### AWS CLI

Per ottenere un elenco di risorse contrassegnate

L'`get-resources` esempio seguente visualizza un elenco di risorse dell'account contrassegnate con il nome e il valore della chiave specificati.

```
aws resourcegroupstaggingapi get-resources \
  --tag-filters Key=Environment,Values=Production \
  --tags-per-page 100
```

Output:

```
{
  "ResourceTagMappingList": [
    {
      "ResourceARN": " arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-7sbz2Kz0",
      "Tags": [
        {
          "Key": "Environment",
          "Value": "Production"
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, vedere [GetResources](#) nel Resource Groups Tagging API Reference.

- Per API i dettagli, vedere [GetResources](#) in AWS CLI Command Reference.

## get-tag-keys

Il seguente esempio di codice mostra come utilizzare `get-tag-keys`.

### AWS CLI

Per ottenere un elenco di tutte le chiavi dei tag

L'`get-tag-keys` esempio seguente recupera l'elenco di tutti i nomi delle chiavi dei tag utilizzati dalle risorse dell'account.

```
aws resourcegroupstaggingapi get-tag-keys
```

Output:

```
{
  "TagKeys": [
    "Environment",
    "CostCenter",
    "Department"
  ]
}
```

Per ulteriori informazioni, vedere [GetTagKeys](#) nel Resource Groups Tagging API Reference.

- Per API i dettagli, vedere [GetTagKeys](#) in AWS CLI Command Reference.

## get-tag-values

Il seguente esempio di codice mostra come utilizzare `get-tag-values`.

### AWS CLI

Per ottenere un elenco di tutti i valori dei tag

L'`get-tag-values` esempio seguente visualizza tutti i valori utilizzati per la chiave specificata per tutte le risorse del

```
aws resourcegroupstaggingapi get-tag-values \
  --key=Environment
```

**Output:**

```
{
  "TagValues": [
    "Alpha",
    "Gamma",
    "Production"
  ]
}
```

Per ulteriori informazioni, vedere [GetTagValues](#) nel Resource Groups Tagging API Reference.

- Per API i dettagli, vedere [GetTagValues](#) in AWS CLI Command Reference.

**tag-resources**

Il seguente esempio di codice mostra come utilizzare `tag-resources`.

**AWS CLI**

Per allegare un tag a una risorsa

L'`tag-resources` esempio seguente contrassegna la risorsa specificata con un nome e un valore chiave.

```
aws resourcegroupstaggingapi tag-resources \
  --resource-arn-list arn:aws:s3:::MyProductionBucket \
  --tags Environment=Production, CostCenter=1234
```

**Output:**

```
{
  "FailedResourcesMap": {}
}
```

Per ulteriori informazioni, vedere [TagResources](#) nel Resource Groups Tagging API Reference.

- Per API i dettagli, vedere [TagResources](#) in AWS CLI Command Reference.

**untag-resources**

Il seguente esempio di codice mostra come utilizzare `untag-resources`.



## AWS CLI

Per rimuovere un tag da una risorsa

L'`untag-resources` esempio seguente rimuove le chiavi dei tag specificate e tutti i valori associati dalla risorsa specificata.

```
aws resourcegroupstaggingapi untag-resources \  
  --resource-arn-list arn:aws:s3:::awsexamplebucket \  
  --tag-keys Environment CostCenter
```

Output:

```
{  
  "FailedResourcesMap": {}  
}
```

Per ulteriori informazioni, vedere [UntagResources](#) nel Resource Groups Tagging API Reference.

- Per API i dettagli, vedere [UntagResources](#) in AWS CLI Command Reference.

## AWS RoboMaker esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS RoboMaker.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, dove puoi trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **batch-describe-simulation-job**

Il seguente esempio di codice mostra come utilizzare `batch-describe-simulation-job`.

## AWS CLI

Per descrivere in batch i lavori di simulazione

L'batch-describe-simulation-jobesempio seguente recupera i dettagli per i tre lavori di simulazione specificati.

Comando:

```
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
```

Output:

```
{
  "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-21k584kiq1oa",
            "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": []
}

```

- Per API i dettagli, vedere [BatchDescribeSimulationJob](#) in AWS CLI Command Reference.

## cancel-simulation-job

Il seguente esempio di codice mostra come utilizzare `cancel-simulation-job`.

### AWS CLI

Per annullare un processo di simulazione

L'esempio seguente annulla il lavoro di simulazione specificato.

```
aws robomaker cancel-simulation-job \  
  --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-66bbb3gpxm8x
```

- Per API i dettagli, vedere [CancelSimulationJob](#) in AWS CLI Command Reference.

## create-deployment-job

Il seguente esempio di codice mostra come utilizzare `create-deployment-job`.

### AWS CLI

Per creare un processo di distribuzione

Questo esempio crea un processo di distribuzione per la flotta MyFleet. Include una variabile di ambiente denominata "ENVIRONMENT". Inoltre, allega un tag chiamato «Regione».

Comando:

```
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
```

Output:

```
{  
  "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"
}
}

```

- Per API i dettagli, vedere [CreateDeploymentJob](#) in AWS CLI Command Reference.

## create-fleet

Il seguente esempio di codice mostra come utilizzare `create-fleet`.

### AWS CLI

Per creare una flotta

Questo esempio crea una flotta. Allega un tag denominato `Region`.

Comando:

```
aws robomaker create-fleet --name MyFleet --tags Region=East
```

Output:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyOtherFleet/1550771394395",
  "name": "MyFleet",
  "createdAt": 1550771394.0,
  "tags": {
    "Region": "East"
  }
}
```

- Per API i dettagli, vedere [CreateFleet](#) in AWS CLI Command Reference.

## create-robot-application-version

Il seguente esempio di codice mostra come utilizzare `create-robot-application-version`.

### AWS CLI

Per creare una versione dell'applicazione robotica

Questo esempio crea una versione dell'applicazione robot.

Comando:

```
aws robomaker create-robot-application-version --application arn:aws:robomaker:us-
west-2:111111111111:robot-application/MyRobotApplication/1551201873931
```

Output:

```
{
  "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": {}
}

```

- Per API i dettagli, vedere [CreateRobotApplicationVersion](#) in AWS CLI Command Reference.

## create-robot-application

Il seguente esempio di codice mostra come utilizzare `create-robot-application`.

### AWS CLI

Per creare un'applicazione robotica

Questo esempio crea un'applicazione robotica.

Comando:

```

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

```

Output:

```

{
  "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": {}
}
```

- Per API i dettagli, vedere [CreateRobotApplication](#) in AWS CLI Command Reference.

## create-robot

Il seguente esempio di codice mostra come utilizzare `create-robot`.

### AWS CLI

Per creare un robot

Questo esempio crea un robot. Utilizza l'ARMHFarchitettura. Inoltre, allega un tag denominato Region.

Comando:

```
aws robomaker create-robot --name MyRobot --architecture ARMHF --greengrass-group-id 0f728a3c-7dbf-4a3e-976d-d16a8360caba --tags Region=East
```

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [CreateRobot](#) in AWS CLI Command Reference.

## create-simulation-application-version

Il seguente esempio di codice mostra come utilizzare `create-simulation-application-version`.

### AWS CLI

Per creare una versione di un'applicazione di simulazione

Questo esempio crea una versione dell'applicazione robotica.

Comando:

```
aws robomaker create-simulation-application-version --  
application arn:aws:robomaker:us-west-2:111111111111:robot-application/  
MySimulationApplication/1551203427605
```

Output:

```
{  
  "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": {}
  }
}

```

- Per API i dettagli, vedere [CreateSimulationApplicationVersion](#) in AWS CLI Command Reference.

## create-simulation-application

Il seguente esempio di codice mostra come utilizzare `create-simulation-application`.

### AWS CLI

Per creare un'applicazione di simulazione

Questo esempio crea un'applicazione di simulazione.

Comando:

```

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

```

Output:

```

{
  "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": {}
}

```

- Per API i dettagli, vedere [CreateSimulationApplication](#) in AWS CLI Command Reference.

## create-simulation-job

Il seguente esempio di codice mostra come utilizzare `create-simulation-job`.

### AWS CLI

Per creare un lavoro di simulazione

Questo esempio crea un lavoro di simulazione. Utilizza un'applicazione robotica e un'applicazione di simulazione.

Comando:

```

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,launc
--tags Region=North

```

Output:

```

{
  "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,
"simulationTimeMillis": 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"
}
}

```

- Per API i dettagli, vedere [CreateSimulationJob](#) in AWS CLI Command Reference.

## delete-fleet

Il seguente esempio di codice mostra come utilizzare `delete-fleet`.

### AWS CLI

Per eliminare una flotta

Questo esempio elimina un parco veicoli.

Comando:

```
aws robomaker delete-fleet --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771394395
```

- Per API i dettagli, vedere [DeleteFleet](#) in AWS CLI Command Reference.

## delete-robot-application

Il seguente esempio di codice mostra come utilizzare `delete-robot-application`.

AWS CLI

Per eliminare un'applicazione robotica

Questo esempio elimina un'applicazione robotica.

Comando:

```
aws robomaker delete-robot-application --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

- Per API i dettagli, vedere [DeleteRobotApplication](#) in AWS CLI Command Reference.

## delete-robot

Il seguente esempio di codice mostra come utilizzare `delete-robot`.

AWS CLI

Per eliminare un robot

Questo esempio elimina un robot.

Comando:

```
aws robomaker delete-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1540829698778
```

- Per API i dettagli, vedere [DeleteRobot](#) in AWS CLI Command Reference.

## delete-simulation-application

Il seguente esempio di codice mostra come utilizzare `delete-simulation-application`.

### AWS CLI

Per eliminare un'applicazione di simulazione

Questo esempio elimina un'applicazione di simulazione.

Comando:

```
aws robomaker delete-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

- Per API i dettagli, vedere [DeleteSimulationApplication](#) in AWS CLI Command Reference.

## deregister-robot

Il seguente esempio di codice mostra come utilizzare `deregister-robot`.

### AWS CLI

Annullare la registrazione di un robot da una flotta

Questo esempio annulla la registrazione di un robot da una flotta.

Comando:

```
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
```

Output:

```
{
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771358907",
  "robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"
}
```

- Per API i dettagli, vedere [DeregisterRobot](#) in AWS CLI Command Reference.



## describe-deployment-job

Il seguente esempio di codice mostra come utilizzare `describe-deployment-job`.

### AWS CLI

Per descrivere un processo di distribuzione

L'esempio seguente recupera i dettagli sul processo di distribuzione specificato.

```
aws robomaker describe-deployment-job \
  --job arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-
xl8qssl6pbcn
```

Output:

```
{
  "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": {}
}

```

- Per API i dettagli, vedere [DescribeDeploymentJob](#) in AWS CLI Command Reference.

## describe-fleet

Il seguente esempio di codice mostra come utilizzare `describe-fleet`.

### AWS CLI

Per descrivere una flotta

L'`describe-fleet` esempio seguente recupera i dettagli per la flotta specificata.

```

aws robomaker describe-fleet \
  --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771358907

```

Output:

```

{
  "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": {}
}
```

- Per API i dettagli, vedere [DescribeFleet](#) in AWS CLI Command Reference.

## describe-robot-application

Il seguente esempio di codice mostra come utilizzare `describe-robot-application`.

### AWS CLI

Per descrivere un'applicazione robotica

Questo esempio descrive un'applicazione robotica.

Comando:

```
aws robomaker describe-robot-application --application arn:aws:robomaker:us-west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

Output:

```
{
  "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": {}
}
```

- Per API i dettagli, vedere [DescribeRobotApplication](#) in AWS CLI Command Reference.

## describe-robot

Il seguente esempio di codice mostra come utilizzare `describe-robot`.

### AWS CLI

Per descrivere un robot

Questo esempio descrive un robot.

Comando:

```
aws robomaker describe-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398
```

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [DescribeRobot](#) in AWS CLI Command Reference.

## describe-simulation-application

Il seguente esempio di codice mostra come utilizzare `describe-simulation-application`.

## AWS CLI

Per descrivere un'applicazione di simulazione

Questo esempio descrive un'applicazione di simulazione.

Comando:

```
aws robomaker describe-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

Output:

```
{
  "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": {}
}
```

- Per API i dettagli, vedere [DescribeSimulationApplication](#) in AWS CLI Command Reference.

## describe-simulation-job

Il seguente esempio di codice mostra come utilizzare `describe-simulation-job`.

### AWS CLI

Per descrivere un lavoro di simulazione

Questo esempio descrive un lavoro di simulazione.

Comando:

```
aws robomaker describe-simulation-job --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-pql32v7pfjy6
```

Output:

```
{
  "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": {}
}

```

- Per API i dettagli, vedere [DescribeSimulationJob](#) in AWS CLI Command Reference.

## list-deployment-jobs

Il seguente esempio di codice mostra come utilizzare `list-deployment-jobs`.

### AWS CLI

Per elencare i lavori di distribuzione

L'`list-deployment-jobs`esempio seguente recupera un elenco di processi di distribuzione.

```
aws robomaker list-deployment-jobs
```

Output:

```

{
  "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
}
]
}

```

- Per API i dettagli, vedere [ListDeploymentJobs](#) in AWS CLI Command Reference.



## list-fleets

Il seguente esempio di codice mostra come utilizzare `list-fleets`.

### AWS CLI

Per elencare le flotte

Questo esempio elenca le flotte. Verranno restituite un massimo di 20 flotte.

Comando:

```
aws robomaker list-fleets --max-items 20
```

Output:

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [ListFleets](#) in AWS CLI Command Reference.

## list-robot-applications

Il seguente esempio di codice mostra come utilizzare `list-robot-applications`.

### AWS CLI

Per elencare le applicazioni robotiche

Questo esempio elenca le applicazioni robotiche. I risultati sono limitati a 20 applicazioni robotiche.

Comando:

```
aws robomaker list-robot-applications --max-results 20
```

Output:

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [ListRobotApplications](#) in AWS CLI Command Reference.

## list-robots

Il seguente esempio di codice mostra come utilizzare `list-robots`.

AWS CLI

Per elencare i robot

Questo esempio elenca i robot. Verranno restituiti un massimo di 20 robot.

Comando:

```
aws robomaker list-robots --max-results 20
```

Output:

```
{
  "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",
      "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"
    }
]
}
```

- Per API i dettagli, vedere [ListRobots](#) in AWS CLI Command Reference.

## list-simulation-applications

Il seguente esempio di codice mostra come utilizzare `list-simulation-applications`.

### AWS CLI

Per elencare le applicazioni di simulazione

Questo esempio elenca le applicazioni di simulazione. Verranno restituite un massimo di 20 applicazioni di simulazione.

Comando:

```
aws robomaker list-simulation-applications --max-results 20
```

Output:

```
{
  "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
  }
]
}

```

- Per API i dettagli, vedere [ListSimulationApplications](#) in AWS CLI Command Reference.

## list-simulation-jobs

Il seguente esempio di codice mostra come utilizzare `list-simulation-jobs`.

### AWS CLI

Per elencare i lavori di simulazione

Questo esempio elenca i lavori di simulazione.

Comando:

**aws robomaker list-simulation-jobs**

## Output:

```
{
  "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-b27c4rkrtzcx",
      "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-51vxjby4q8t",
      "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-
vw8lvh061nqt",
    "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-
txy5ypxmh84",
    "lastUpdatedAt": 1543437488.0,
    "status": "Completed",
    "simulationApplicationNames": [
      "AWSRoboMakerHelloWorld-1543437372341__yb_Jg961"
    ],
    "robotApplicationNames": [
      "AWSRoboMakerHelloWorld-1543437372341_lNbmKHvs9"
    ]
  }
]
}

```

- Per API i dettagli, vedere [ListSimulationJobs](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

## AWS CLI

Per elencare i tag di una risorsa

Questo esempio elenca i tag di una AWS RoboMaker risorsa.

Comando:

```
aws robomaker list-tags-for-resource --resource-arn "arn:aws:robomaker:us-west-2:111111111111:robot/Robby_the_Robot/1544035373264"
```

Output:

```
{
  "tags": {
    "Region": "North",
    "Stage": "Initial"
  }
}
```

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## register-robot

Il seguente esempio di codice mostra come utilizzare `register-robot`.

## AWS CLI

Per registrare un robot

Questo esempio registra un robot in una flotta.

Comando:

```
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
```

Output:

```
{
```



```
"fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/MyFleet/1550771358907",  
"robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"  
}
```

- Per API i dettagli, vedere [RegisterRobot](#) in AWS CLI Command Reference.

## restart-simulation-job

Il seguente esempio di codice mostra come utilizzare `restart-simulation-job`.

### AWS CLI

Per riavviare una simulazione

Questo esempio riavvia una simulazione.

Comando:

```
aws robomaker restart-simulation-job --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-t6rdgt70mftr
```

- Per API i dettagli, vedere [RestartSimulationJob](#) in AWS CLI Command Reference.

## sync-deployment-job

Il seguente esempio di codice mostra come utilizzare `sync-deployment-job`.

### AWS CLI

Per sincronizzare un processo di distribuzione

Questo esempio sincronizza un processo di distribuzione.

Comando:

```
aws robomaker sync-deployment-job --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/Trek/1539894765711
```

Output:

```
{
  "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
}
```

- Per API i dettagli, vedere [SyncDeploymentJob](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per etichettare una risorsa

Questo esempio contrassegna una risorsa. Allega due tag: Region e Stage.

Comando:

```
aws robomaker tag-resource --resource-arn "arn:aws:robomaker:us-
west-2:111111111111:robot/MyRobot/1544035373264" --tags Region=North,Stage=Initial
```

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una risorsa

Questo esempio rimuove un tag da una risorsa. Rimuove il tag `Region`.

Comando:

```
aws robomaker untag-resource --resource-arn "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1544035373264" --tag-keys Region
```

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-robot-application

Il seguente esempio di codice mostra come utilizzare `update-robot-application`.

### AWS CLI

Per aggiornare un'applicazione robotica

Questo esempio aggiorna un'applicazione robotica.

Comando:

```
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
```

Output:

```
{
  "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"
}

```

- Per API i dettagli, vedere [UpdateRobotApplication](#) in AWS CLI Command Reference.

## update-simulation-application

Il seguente esempio di codice mostra come utilizzare `update-simulation-application`.

### AWS CLI

Per aggiornare un'applicazione di simulazione

Questo esempio aggiorna un'applicazione di simulazione.

Comando:

```

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

```

Output:

```

{
  "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"
}
```

- Per API i dettagli, vedere [UpdateSimulationApplication](#) in AWS CLI Command Reference.

## Esempi di Route 53 che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Route 53.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### change-resource-record-sets

Il seguente esempio di codice mostra come utilizzare `change-resource-record-sets`.

#### AWS CLI

Per creare, aggiornare o eliminare un set di record di risorse

Il `change-resource-record-sets` comando seguente crea un set di record di risorse utilizzando la configurazione in JSON formato `hosted-zone-id` `Z1R8UBAEXAMPLE` e nel file: `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
```

Per ulteriori informazioni, POST `ChangeResourceRecordSets` consulta [Amazon Route 53 API Reference](#).

La configurazione nel JSON file dipende dal tipo di set di record di risorse che desideri creare:

`BasicWeightedAliasWeighted AliasLatencyLatency AliasFailoverFailover Alias`

Sintassi di base:

```
{
  "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"
          },
          {...}
        ]
      }
    },
    ...
  ],
}
```

```

    {...}
  ]
}

```

### Sintassi ponderata:

```

{
  "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"
      }
    },
    {...}
  ]
}

```

### Sintassi degli 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",
        "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"
}
},
{...}
]
}

```

### Sintassi degli alias ponderati:

```

{
  "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"
      }
    },
    {...}
  ]
}

```

### Sintassi della latenza:

```

{
  "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"
    }
  },
  {...}
]
}

```

### Sintassi dell'alias di latenza:

```

{
  "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"
      }
    }
  ]
}

```

```

    }
  },
  {...}
]
}

```

### Sintassi di failover:

```

{
  "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"
      }
    },
    {...}
  ]
}

```

### Sintassi degli alias di failover:

```

{
  "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"
  }
},
{...}
]
}

```

- Per API i dettagli, vedere [ChangeResourceRecordSets](#) in AWS CLI Command Reference.

## change-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `change-tags-for-resource`.

### AWS CLI

Il comando seguente aggiunge un tag denominato `owner` a una risorsa `healthcheck` specificata da ID:

```
aws route53 change-tags-for-resource --resource-type healthcheck --resource-id 6233434j-18c1-34433-ba8e-3443434 --add-tags Key=owner,Value=myboss
```

Il comando seguente rimuove un tag denominato `owner` da una risorsa della zona ospitata specificata da ID:

```
aws route53 change-tags-for-resource --resource-type hostedzone --resource-id Z1523434445 --remove-tag-keys owner
```

- Per API i dettagli, vedere [ChangeTagsForResource](#) in AWS CLI Command Reference.

## create-health-check

Il seguente esempio di codice mostra come utilizzare `create-health-check`.

## AWS CLI

Per creare un controllo sanitario

Il `create-health-check` comando seguente crea un controllo dello stato utilizzando il riferimento al chiamante `2014-04-01-18:47` e la configurazione JSON -formatted nel file: `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
```

JSONsintassi:

```
{
  "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
}
```

Per aggiungere il controllo dello stato di salute a un set di record di risorse Route 53, utilizzare il `change-resource-record-sets` comando.

Per ulteriori informazioni, consulta [Amazon Route 53 Health Checks and DNS Failover](#) nella [Amazon Route 53 Developer Guide](#).

- Per API i dettagli, consulta [CreateHealthCheck AWS CLI Command Reference](#).

## create-hosted-zone

Il seguente esempio di codice mostra come utilizzare `create-hosted-zone`.

## AWS CLI

Per creare una zona ospitata

Il `create-hosted-zone` comando seguente aggiunge una zona ospitata denominata `example.com` utilizzando il riferimento del chiamante. `2014-04-01-18:47` Il commento opzionale include uno spazio, quindi deve essere racchiuso tra virgolette:

```
aws route53 create-hosted-zone --name example.com --caller-reference 2014-04-01-18:47 --hosted-zone-config Comment="command-line version"
```

Per ulteriori informazioni, consulta [Working with Hosted Zones](#) nella [Amazon Route 53 Developer Guide](#).

- Per API i dettagli, consulta [CreateHostedZone AWS CLI Command Reference](#).

## delete-health-check

Il seguente esempio di codice mostra come utilizzare `delete-health-check`.

### AWS CLI

Per eliminare un controllo sanitario

Il `delete-health-check` comando seguente elimina il controllo dello stato con un `health-check-id` `die75b48d9-547a-4c3d-88a5-ae4002397608`:

```
aws route53 delete-health-check --health-check-id e75b48d9-547a-4c3d-88a5-ae4002397608
```

- Per API i dettagli, vedere [DeleteHealthCheck](#) in [AWS CLI Command Reference](#).

## delete-hosted-zone

Il seguente esempio di codice mostra come utilizzare `delete-hosted-zone`.

### AWS CLI

Per eliminare una zona ospitata

Il `delete-hosted-zone` comando seguente elimina la zona ospitata con un `id` `diZ36KTIQEXAMPLE`:

```
aws route53 delete-hosted-zone --id Z36KTIQEXAMPLE
```

- Per API i dettagli, vedere [DeleteHostedZone](#) in AWS CLI Command Reference.

## get-change

Il seguente esempio di codice mostra come utilizzare `get-change`.

### AWS CLI

Per ottenere lo stato di una modifica ai set di record di risorse

Il `get-change` comando seguente ottiene lo stato e altre informazioni sulla `change-resource-record-sets` richiesta che presenta un Id di `/change/CWPIK4URU2I5S`:

```
aws route53 get-change --id /change/CWPIK4URU2I5S
```

- Per API i dettagli, vedere [GetChange](#) in AWS CLI Command Reference.

## get-health-check

Il seguente esempio di codice mostra come utilizzare `get-health-check`.

### AWS CLI

Per ottenere informazioni su un controllo sanitario

Il `get-health-check` comando seguente consente di ottenere informazioni sul controllo dello stato `health-check-id` di salute che include `02ec8401-9879-4259-91fa-04e66d094674`:

```
aws route53 get-health-check --health-check-id 02ec8401-9879-4259-91fa-04e66d094674
```

- Per API i dettagli, vedere [GetHealthCheck](#) in AWS CLI Command Reference.

## get-hosted-zone

Il seguente esempio di codice mostra come utilizzare `get-hosted-zone`.

### AWS CLI

Per ottenere informazioni su una zona ospitata

Il `get-hosted-zone` comando seguente ottiene informazioni sulla zona ospitata con un `id` di `Z1R8UBAEXAMPLE`:

```
aws route53 get-hosted-zone --id Z1R8UBAEXAMPLE
```

- Per API i dettagli, vedere [GetHostedZone](#) in AWS CLI Command Reference.

## **list-health-checks**

Il seguente esempio di codice mostra come utilizzare `list-health-checks`.

### AWS CLI

Per elencare i controlli sanitari associati all' AWS account corrente

Il `list-health-checks` comando seguente elenca informazioni dettagliate sui primi 100 controlli sanitari associati all' AWS account corrente. :

```
aws route53 list-health-checks
```

Se hai più di 100 controlli sanitari o se desideri elencarli in gruppi più piccoli di 100, includi il `--max-items` parametro. Ad esempio, per elencare i controlli sanitari uno alla volta, utilizzate il comando seguente:

```
aws route53 list-health-checks --max-items 1
```

Per visualizzare il prossimo controllo di integrità, prendi il valore di `NextToken` dalla risposta al comando precedente e includilo nel `--starting-token` parametro, ad esempio:

```
aws route53 list-health-checks --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- Per API i dettagli, vedere [ListHealthChecks](#) in AWS CLI Command Reference.

## **list-hosted-zones-by-name**

Il seguente esempio di codice mostra come utilizzare `list-hosted-zones-by-name`.

### AWS CLI

Il comando seguente elenca fino a 100 zone ospitate ordinate per nome di dominio:

```
aws route53 list-hosted-zones-by-name
```

Output:

```
{
  "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"
}
```

Il comando seguente elenca le zone ospitate ordinate per nome, a partire da `www.example.com`:

```
aws route53 list-hosted-zones-by-name --dns-name www.example.com
```

Output:

```
{
  "HostedZones": [
    {
      "ResourceRecordSetCount": 2,
```



```
    "CallerReference": "mwunder120150527-1",
    "Config": {
      "Comment": "test",
      "PrivateZone": false
    },
    "Id": "/hostedzone/Z3P5QSUBK4P0TI",
    "Name": "www.example.com."
  }
],
"DNSName": "www.example.com",
"IsTruncated": false,
"MaxItems": "100"
}
```

- Per API i dettagli, vedere [ListHostedZonesByName](#) in AWS CLI Command Reference.

## list-hosted-zones

Il seguente esempio di codice mostra come utilizzare `list-hosted-zones`.

### AWS CLI

Per elencare le zone ospitate associate all' AWS account corrente

Il `list-hosted-zones` comando seguente elenca le informazioni di riepilogo sulle prime 100 zone ospitate associate all' AWS account corrente. :

```
aws route53 list-hosted-zones
```

Se disponi di oltre 100 zone ospitate o se desideri elencarle in gruppi più piccoli di 100, includi il parametro `--max-items`. Ad esempio, per elencare le zone ospitate una alla volta, utilizza il comando seguente:

```
aws route53 list-hosted-zones --max-items 1
```

Per visualizzare le informazioni sulla zona ospitata successiva, prendi il valore di `NextToken` dalla risposta al comando precedente e includilo nel parametro `--starting-token`, ad esempio:

```
aws route53 list-hosted-zones --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- Per API i dettagli, vedere [ListHostedZones](#) in AWS CLI Command Reference.

## list-query-logging-configs

Il seguente esempio di codice mostra come utilizzare `list-query-logging-configs`.

### AWS CLI

Per elencare le configurazioni di registrazione delle query

L'`list-query-logging-configs` seguente elenca le informazioni sulle prime 100 configurazioni di registrazione delle query nell' AWS account, per la zona ospitata.

Z10X3WQEXAMPLE

```
aws route53 list-query-logging-configs \  
  --hosted-zone-id Z10X3WQEXAMPLE
```

Output:

```
{  
  "QueryLoggingConfigs": [  
    {  
      "Id": "964ff34e-ae03-4f06-80a2-9683cexample",  
      "HostedZoneId": "Z10X3WQEXAMPLE",  
      "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:111122223333:log-  
group:/aws/route53/example.com:*"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta la sezione [Logging DNS delle query](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListQueryLoggingConfigs](#) Reference.

## list-resource-record-sets

Il seguente esempio di codice mostra come utilizzare `list-resource-record-sets`.

### AWS CLI

Per elencare i set di record di risorse in una zona ospitata

Il `list-resource-record-sets` comando seguente elenca informazioni di riepilogo sui primi 100 set di record di risorse in una zona ospitata specificata. :

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE
```

Se la zona ospitata contiene più di 100 set di record di risorse o se desideri elencarli in gruppi più piccoli di 100, includi il `--max-items` parametro. Ad esempio, per elencare i set di record di risorse uno alla volta, utilizzate il comando seguente:

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1
```

Per visualizzare le informazioni sul prossimo record di risorse impostato nella zona ospitata, prendete il valore di `NextToken` dalla risposta al comando precedente e includetelo nel `--starting-token` parametro, ad esempio:

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1  
--starting-token Z3M3LMPEXAMPLE
```

Per visualizzare tutti i set di record di risorse con un nome particolare, utilizzate il `--query` parametro per filtrarli. Per esempio:

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --  
query "ResourceRecordSets[?Name == 'example.domain.']"
```

- Per API i dettagli, vedere [ListResourceRecordSets](#) in AWS CLI Command Reference.

## Esempi di registrazione del dominio Route 53 utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando la registrazione del dominio AWS Command Line Interface with Route 53.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **check-domain-availability**

Il seguente esempio di codice mostra come utilizzare `check-domain-availability`.

#### AWS CLI

Per determinare se è possibile registrare un nome di dominio con Route 53

Il `check-domain-availability` comando seguente restituisce informazioni sulla disponibilità del nome `example.com` di dominio per la registrazione tramite Route 53.

Questo comando viene eseguito solo nella `us-east-1` regione. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains check-domain-availability \  
  --region us-east-1 \  
  --domain-name example.com
```

Output:

```
{  
  "Availability": "UNAVAILABLE"  
}
```

Route 53 supporta un gran numero di domini di primo livello (TLDs), come `.com` e `.jp`, ma non supportiamo tutti i domini disponibili. TLDs Se si verifica la disponibilità di un dominio e Route 53 non supporta il TLD, `check-domain-availability` restituisce il seguente messaggio.

```
An error occurred (UnsupportedTLD) when calling the CheckDomainAvailability  
operation: <top-level domain> tld is not supported.
```

Per un elenco dei [domini TLDs che puoi utilizzare per registrare un dominio con Route 53](#), consulta [Domini che puoi registrare con Amazon Route 53 nella Amazon Route 53 Developer Guide](#). Per ulteriori informazioni sulla registrazione di domini con Amazon Route 53, consulta [Registrazione di un nuovo dominio](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta Command [CheckDomainAvailability](#) Reference AWS CLI .

## check-domain-transferability

Il seguente esempio di codice mostra come utilizzare `check-domain-transferability`.

### AWS CLI

Per determinare se un dominio può essere trasferito su Route 53

Il `check-domain-transferability` comando seguente restituisce informazioni sulla possibilità di trasferire il nome `example.com` di dominio su Route 53.

Questo comando viene eseguito solo nella `us-east-1` regione. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains check-domain-transferability \
  --region us-east-1 \
  --domain-name example.com
```

Output:

```
{
  "Transferability": {
    "Transferable": "UNTRANSFERABLE"
  }
}
```

Per ulteriori informazioni, consulta [Trasferimento della registrazione di un dominio su Amazon Route 53](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [CheckDomainTransferability AWS CLI Command Reference](#).

## delete-tags-for-domain

Il seguente esempio di codice mostra come utilizzare `delete-tags-for-domain`.

### AWS CLI

Per eliminare i tag per un dominio

Il `delete-tags-for-domain` comando seguente elimina tre tag dal dominio specificato. Si noti che si specifica solo la chiave del tag, non il valore del tag.

Questo comando viene eseguito solo nella `us-east-1` regione. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains delete-tags-for-domain \  
  --region us-east-1 \  
  --domain-name example.com \  
  --tags-to-delete accounting-key hr-key engineering-key
```

Questo comando non produce alcun output.

Per confermare che i tag sono stati eliminati, puoi eseguire [list-tags-for-domain](#). Per ulteriori informazioni, consulta [Tagging Amazon Route 53 Resources](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [DeleteTagsForDomain AWS CLI Command Reference](#).

## **disable-domain-auto-renew**

Il seguente esempio di codice mostra come utilizzare `disable-domain-auto-renew`.

### AWS CLI

Per disabilitare il rinnovo automatico di un dominio

Il `disable-domain-auto-renew` comando seguente configura Route 53 in modo che non rinnovi automaticamente il dominio `example.com` prima della scadenza della registrazione per il dominio.

Questo comando viene eseguito solo nella regione `us-east-1`. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains disable-domain-auto-renew \  
  --region us-east-1 \  
  --domain-name example.com
```

Questo comando non produce alcun output.

Per confermare che l'impostazione è stata modificata, puoi eseguire [get-domain-detail](#).

Se il rinnovo automatico è disabilitato, il valore di `AutoRenew` è `False`. Per ulteriori informazioni sul rinnovo automatico, consulta [Renewing Registration for a Domain < https://](#)

[docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [DisableDomainAutoRenew AWS CLI Command Reference](#).

## **disable-domain-transfer-lock**

Il seguente esempio di codice mostra come utilizzare `disable-domain-transfer-lock`.

### AWS CLI

Per disabilitare il blocco del trasferimento su un dominio

Il `disable-domain-transfer-lock` comando seguente rimuove il blocco di trasferimento sul dominio `example.com` in modo che il dominio possa essere trasferito a un altro registrar. Questo comando modifica lo `clientTransferProhibited` stato.

Questo comando viene eseguito solo nella `us-east-1` regione. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains disable-domain-transfer-lock \  
  --region us-east-1 \  
  --domain-name example.com
```

Output:

```
{  
  "OperationId": "3f28e0ac-126a-4113-9048-cc930example"  
}
```

Per confermare che il blocco del trasferimento è stato modificato, puoi eseguire [get-domain-detail](#). Quando il blocco del trasferimento è disabilitato, il valore di `StatusList` non include `clientTransferProhibited`.

Per ulteriori informazioni sul processo di trasferimento, consulta [Trasferimento di un dominio da Amazon Route 53 a un altro registrar](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DisableDomainTransferLock Reference](#).

## **enable-domain-auto-renew**

Il seguente esempio di codice mostra come utilizzare `enable-domain-auto-renew`.

## AWS CLI

Per abilitare il rinnovo automatico di un dominio

Il `enable-domain-auto-renew` comando seguente configura Route 53 per rinnovare automaticamente il dominio `example.com` prima della scadenza della registrazione per il dominio.

Questo comando viene eseguito solo nella regione `us-east-1`. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains enable-domain-auto-renew \  
  --region us-east-1 \  
  --domain-name example.com
```

Questo comando non produce alcun output. Per confermare che l'impostazione è stata modificata, puoi eseguire [get-domain-detail](#). Se il rinnovo automatico è abilitato, il valore di `AutoRenew` è `True`.

Per ulteriori informazioni sul rinnovo automatico, consulta [Renewing Registration for a Domain < https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html](https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [EnableDomainAutoRenew AWS CLI Command Reference](#).

## **enable-domain-transfer-lock**

Il seguente esempio di codice mostra come utilizzare `enable-domain-transfer-lock`.

## AWS CLI

Per abilitare il blocco del trasferimento su un dominio

Il `enable-domain-transfer-lock` comando seguente blocca il dominio specificato in modo che non possa essere trasferito a un altro registrar. Questo comando modifica lo `clientTransferProhibited` stato.

Questo comando viene eseguito solo nella `us-east-1` regione. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains enable-domain-transfer-lock \  
  --region us-east-1 \  
  --domain-name example.com
```



```
--domain-name example.com
```

Output:

```
{  
  "OperationId": "3f28e0ac-126a-4113-9048-cc930example"  
}
```

Per confermare che il blocco del trasferimento è stato modificato, puoi eseguire [get-domain-detail](#). Quando il blocco del trasferimento è abilitato, il valore di `StatusList` include `clientTransferProhibited`.

Per ulteriori informazioni sul processo di trasferimento, consulta [Trasferimento di un dominio da Amazon Route 53 a un altro registrar](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [EnableDomainTransferLockReference](#).

## get-contact-reachability-status

Il seguente esempio di codice mostra come utilizzare `get-contact-reachability-status`.

AWS CLI

Per determinare se il contatto del registrante ha risposto a un'e-mail di conferma

Il `get-contact-reachability-status` comando seguente restituisce informazioni sul fatto che il contatto del registrante per il dominio specificato abbia risposto a un'e-mail di conferma.

Questo comando viene eseguito solo nella regione `us-east-1`. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains get-contact-reachability-status \  
  --region us-east-1 \  
  --domain-name example.com
```

Output:

```
{  
  "domainName": "example.com",  
  "status": "DONE"  
}
```

Per ulteriori informazioni, consulta [Reinvio di e-mail di autorizzazione e conferma](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [GetContactReachabilityStatus AWS CLI Command Reference](#).

## get-domain-detail

Il seguente esempio di codice mostra come utilizzare `get-domain-detail`.

### AWS CLI

Per ottenere informazioni dettagliate su un dominio specifico

Il `get-domain-detail` comando seguente visualizza informazioni dettagliate sul dominio specificato.

Questo comando viene eseguito solo nella `us-east-1` regione. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains get-domain-detail \
  --region us-east-1 \
  --domain-name example.com
```

Output:

```
{
  "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"
]
}

```

- Per API i dettagli, vedere [GetDomainDetail](#) in AWS CLI Command Reference.

## get-domain-suggestions

Il seguente esempio di codice mostra come utilizzare `get-domain-suggestions`.

### AWS CLI

Per ottenere un elenco dei nomi di dominio suggeriti

Il `get-domain-suggestions` comando seguente visualizza un elenco di nomi di dominio suggeriti in base al nome di dominio `example.com`. La risposta include solo i nomi di dominio disponibili. Questo comando viene eseguito solo nella `us-east-1` regione. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```

aws route53domains get-domain-suggestions \
  --region us-east-1 \
  --domain-name example.com \
  --suggestion-count 10 \
  --only-available

```

Output:

```

{
  "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"
  }
]
}
```

- Per API i dettagli, vedere [GetDomainSuggestions](#) in AWS CLI Command Reference.

## get-operation-detail

Il seguente esempio di codice mostra come utilizzare `get-operation-detail`.

## AWS CLI

Per ottenere lo stato corrente di un'operazione

Alcune operazioni di registrazione del dominio funzionano in modo asincrono e restituiscono una risposta prima del termine. Queste operazioni restituiscono un ID operativo che è possibile utilizzare per ottenere lo stato corrente. Il `get-operation-detail` comando seguente restituisce lo stato dell'operazione specificata.

Questo comando viene eseguito solo nella `us-east-1` regione. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains get-operation-detail \  
  --region us-east-1 \  
  --operation-id edbd8d63-7fe7-4343-9bc5-54033example
```

Output:

```
{  
  "OperationId": "edbd8d63-7fe7-4343-9bc5-54033example",  
  "Status": "SUCCESSFUL",  
  "DomainName": "example.com",  
  "Type": "DOMAIN_LOCK",  
  "SubmittedDate": 1573749367.864  
}
```

- Per API i dettagli, vedere [GetOperationDetail](#) in AWS CLI Command Reference.

## list-domains

Il seguente esempio di codice mostra come utilizzare `list-domains`.

### AWS CLI

Per elencare i domini registrati con l'account corrente AWS

Il `list-domains` comando seguente elenca informazioni di riepilogo sui domini registrati con l'account corrente AWS .

Questo comando viene eseguito solo nella `us-east-1` regione. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains list-domains
--region us-east-1
```

Output:

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [ListDomains](#) in AWS CLI Command Reference.

## list-operations

Il seguente esempio di codice mostra come utilizzare `list-operations`.

### AWS CLI

Per elencare lo stato delle operazioni che restituiscono un ID di operazione

Alcune operazioni di registrazione del dominio vengono eseguite in modo asincrono e restituiscono una risposta prima del termine. Queste operazioni restituiscono un ID di operazione che è possibile utilizzare per ottenere lo stato corrente. Il `list-operations` comando seguente

elenca informazioni di riepilogo, incluso lo stato, sulle operazioni correnti di registrazione del dominio.

Questo comando viene eseguito solo nella regione `us-east-1`. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains list-operations
--region us-east-1
```

Output:

```
{
  "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
    }
  ]
}
```

L'output include tutte le operazioni che restituiscono un ID di operazione e che hai eseguito su tutti i domini che hai mai registrato utilizzando l'account corrente AWS . Se desideri ottenere solo le



operazioni inviate dopo una data specificata, puoi includere il `submitted-since` parametro e specificare una data in formato Unix e Coordinated Universal Time (UTC). Il comando seguente ottiene lo stato di tutte le operazioni inviate dopo le 12:00 del UTC 1° gennaio 2020.

```
aws route53domains list-operations \  
  --submitted-since 1577836800
```

- Per API i dettagli, vedere [ListOperations](#) in AWS CLI Command Reference.

## list-tags-for-domain

Il seguente esempio di codice mostra come utilizzare `list-tags-for-domain`.

### AWS CLI

Per elencare i tag di un dominio

Il `list-tags-for-domain` comando seguente elenca i tag attualmente associati al dominio specificato.

Questo comando viene eseguito solo nella `us-east-1` regione. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains list-tags-for-domain \  
  --region us-east-1 \  
  --domain-name example.com
```

Output:

```
{  
  "TagList": [  
    {  
      "Key": "key1",  
      "Value": "value1"  
    },  
    {  
      "Key": "key2",  
      "Value": "value2"  
    }  
  ]  
}
```

```
}
```

Per ulteriori informazioni, consulta [Tagging Amazon Route 53 Resources](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [ListTagsForDomain AWS CLI Command Reference](#).

## register-domain

Il seguente esempio di codice mostra come utilizzare `register-domain`.

### AWS CLI

Per registrare un dominio

Il `register-domain` comando seguente registra un dominio, recuperando tutti i valori dei parametri da un JSON file in formato.

Questo comando viene eseguito solo nella regione `us-east-1`. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains register-domain \  
  --region us-east-1 \  
  --cli-input-json file://register-domain.json
```

Contenuto di `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
}
```

### Output:

```
{
  "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

Per confermare che l'operazione è riuscita, puoi eseguire `get-operation-detail`. Per ulteriori informazioni, vedere [get-operation-detail](#).

Per maggiori informazioni, consulta [Registrazione di un nuovo dominio](#) nella Guida per gli sviluppatori Amazon Route 53.

Per informazioni su quali domini di primo livello (TLDs) richiedono valori `ExtraParams` e quali sono i valori validi, consulta [ExtraParam](#) Amazon Route 53 API Reference.

- Per API i dettagli, consulta AWS CLI Command [RegisterDomain](#) Reference.

## renew-domain

Il seguente esempio di codice mostra come utilizzare `renew-domain`.

### AWS CLI

Per rinnovare un dominio

Il `renew-domain` comando seguente rinnova il dominio specificato per cinque anni. Per ottenere il valore `current-expiry-year`, usate il `get-domain-detail` comando e convertite il valore di `ExpirationDate` dal formato Unix.

Questo comando viene eseguito solo nella `us-east-1` regione. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains renew-domain \  
  --region us-east-1 \  
  --domain-name example.com \  
  --duration-in-years 5 \  
  --current-expiry-year 2020
```

Output:

```
{  
  "OperationId": "3f28e0ac-126a-4113-9048-cc930example"  
}
```

Per confermare che l'operazione è riuscita, puoi eseguire `get-operation-detail`. Per ulteriori informazioni, vedere [get-operation-detail](#).

Il registro per ogni dominio di primo livello (TLD), ad esempio `.com` o `.org`, controlla il numero massimo di anni per i quali è possibile rinnovare un dominio. Per ottenere il periodo di rinnovo massimo per il tuo dominio, consulta la sezione «Periodo di registrazione e rinnovo» relativa ai [domini che puoi registrare con Amazon Route 53 nella Amazon Route 53 Developer Guide](#). TLD

Per ulteriori informazioni, consulta [Rinnovo della registrazione di un dominio](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [RenewDomain AWS CLI](#) Command Reference.

## resend-contact-reachability-email

Il seguente esempio di codice mostra come utilizzare `resend-contact-reachability-email`.

### AWS CLI

Per inviare nuovamente l'e-mail di conferma all'indirizzo e-mail corrente del registrante, contatta

Il `resend-contact-reachability-email` comando seguente invia nuovamente l'e-mail di conferma all'indirizzo e-mail corrente del contatto del registrante per il dominio `example.com`.

Questo comando viene eseguito solo nella regione `us-east-1`. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains resend-contact-reachability-email \  
  --region us-east-1 \  
  --domain-name example.com
```

Output:

```
{  
  "domainName": "example.com",  
  "emailAddress": "moliveira@example.com",  
  "isAlreadyVerified": true  
}
```

Se il valore di `isAlreadyVerified` è `true`, come in questo esempio, il contatto del registrante ha già confermato che l'indirizzo email specificato è raggiungibile.

Per ulteriori informazioni, consulta [Reinvio di e-mail di autorizzazione e conferma](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [ResendContactReachabilityEmail AWS CLI](#) Command Reference.

## retrieve-domain-auth-code

Il seguente esempio di codice mostra come utilizzare `retrieve-domain-auth-code`.

## AWS CLI

Per ottenere il codice di autorizzazione per un dominio in modo da poterlo trasferire a un altro registrar

Il `retrieve-domain-auth-code` comando seguente ottiene il codice di autorizzazione corrente per il dominio `example.com`. Assegna questo valore a un altro registrar di domini quando desideri trasferire il dominio a quel registrar.

Questo comando viene eseguito solo nella regione `us-east-1`. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains retrieve-domain-auth-code \  
  --region us-east-1 \  
  --domain-name example.com
```

Output:

```
{  
  "AuthCode": ")o!v3dJeXampLe"  
}
```

Per ulteriori informazioni, consulta [Trasferimento di un dominio da Amazon Route 53 a un altro registrar](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [RetrieveDomainAuthCode](#) Reference.

## **transfer-domain**

Il seguente esempio di codice mostra come utilizzare `transfer-domain`.

### AWS CLI

Per trasferire un dominio su Amazon Route 53

Il `transfer-domain` comando seguente trasferisce un dominio su Route 53, con i parametri forniti dal file JSON `-formatted`. `C:\temp\transfer-domain.json`

Questo comando viene eseguito solo nella regione `us-east-1`. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains transfer-domain \  
--region us-east-1 \  
--cli-input-json file://C:\temp\transfer-domain.json
```

Contenuto di `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
}
```

Output:

```
{
  "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

Per confermare che l'operazione è riuscita, puoi eseguire `get-operation-detail`. Per ulteriori informazioni, vedere [get-operation-detail](#).

Per ulteriori informazioni, consulta [Trasferimento della registrazione di un dominio su Amazon Route 53](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [TransferDomain AWS CLI Command Reference](#).

## update-domain-contact-privacy

Il seguente esempio di codice mostra come utilizzare `update-domain-contact-privacy`.



## AWS CLI

Per aggiornare le impostazioni sulla privacy per i contatti di un dominio

Il `update-domain-contact-privacy` comando seguente disattiva la protezione della privacy per il contatto amministrativo del dominio `example.com`. Questo comando viene eseguito solo nella regione `us-east-1`.

Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains update-domain-contact-privacy \  
  --region us-east-1 \  
  --domain-name example.com \  
  --no-admin-privacy
```

Output:

```
{  
  "OperationId": "b3a219e9-d801-4244-b533-b7256example"  
}
```

Per confermare che l'operazione è riuscita, puoi eseguire `get-operation-detail`. Per ulteriori informazioni, vedere [get-operation-detail](#).

Per ulteriori informazioni, consulta [Attivazione o disattivazione della protezione della privacy per le informazioni di contatto per un dominio](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [UpdateDomainContactPrivacy AWS CLI Command Reference](#).

## update-domain-contact

Il seguente esempio di codice mostra come utilizzare `update-domain-contact`.

### AWS CLI

Per aggiornare le informazioni di contatto per un dominio

Il `update-domain-contact` comando seguente aggiorna le informazioni di contatto per un dominio, ottenendo i parametri dal file JSON `-formatted`. `C:\temp\update-domain-contact.json`

Questo comando viene eseguito solo nella regione `us-east-1`. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains update-domain-contact \  
--region us-east-1 \  
--cli-input-json file://C:\temp\update-domain-contact.json
```

Contenuto di `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"
}
}
```

Output:

```
{
  "OperationId": "b3a219e9-d801-4244-b533-b7256example"
}
```

Per confermare che l'operazione è riuscita, puoi eseguire [get-domain-detail](#). Per ulteriori informazioni, consulta [Aggiornamento delle informazioni di contatto per un dominio](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [UpdateDomainContact AWS CLI Command Reference](#).

## update-domain-nameservers

Il seguente esempio di codice mostra come utilizzare `update-domain-nameservers`.

### AWS CLI

Per aggiornare i name server per un dominio

Il `update-domain-nameservers` comando seguente aggiorna i name server per un dominio.

Questo comando viene eseguito solo nella `us-east-1` regione. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
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=
```

Output:

```
{
  "OperationId": "f1691ec4-0e7a-489e-82e0-b19d3example"
}
```

Per confermare che l'operazione è riuscita, puoi eseguire. [get-domain-detail](#)

Per ulteriori informazioni, consulta [Aggiungere o modificare i name server e i record Glue per un dominio](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [UpdateDomainNameservers AWS CLI Command Reference](#).

## update-tags-for-domain

Il seguente esempio di codice mostra come utilizzare `update-tags-for-domain`.

### AWS CLI

Per aggiungere o aggiornare i tag per un dominio

Il `update-tags-for-domain` comando seguente aggiunge o aggiorna due chiavi e i valori corrispondenti per il dominio `example.com`. Per aggiornare il valore di una chiave, è sufficiente includere la chiave e il nuovo valore. Puoi aggiungere o aggiornare i tag in un solo dominio alla volta.

Questo comando viene eseguito solo nella `us-east-1` regione. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
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"
```

Questo comando non produce alcun output. Per confermare che i tag sono stati aggiunti o aggiornati, puoi eseguire [list-tags-for-domain](#).

Per ulteriori informazioni, consulta [Tagging Amazon Route 53 Resources](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [UpdateTagsForDomain AWS CLI Command Reference](#).

## view-billing

Il seguente esempio di codice mostra come utilizzare `view-billing`.

### AWS CLI

Per ottenere informazioni di fatturazione per gli addebiti di registrazione del dominio per l'account corrente AWS

Il `view-billing` comando seguente restituisce tutti i record di fatturazione relativi al dominio per l'account corrente per il periodo compreso tra il 1° gennaio 2018 (1514764800 nel fuso orario Unix) e la mezzanotte del 31 dicembre 2019 (1577836800 nel fuso orario Unix).

Questo comando viene eseguito solo nella regione `us-east-1`. Se la regione predefinita è impostata su `us-east-1`, è possibile omettere il `region` parametro.

```
aws route53domains view-billing \  
  --region us-east-1 \  
  --start-time 1514764800 \  
  --end-time 1577836800
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

Per ulteriori informazioni, [ViewBilling](#) consulta Amazon Route 53 API Reference.

- Per API i dettagli, consulta [ViewBilling AWS CLI](#) Command Reference.

## Esempi di profili Route 53 utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando i profili AWS Command Line Interface with Route 53.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **associate-profile**

Il seguente esempio di codice mostra come utilizzare `associate-profile`.

AWS CLI

Per associare un profilo

L'`associate-profile` esempio seguente associa un profilo a un VPC.

```
aws route53profiles associate-profile \  
  --name test-association \  
  --profile-id rp-4987774726example \  
  --resource-id vpc-0af3b96b3example
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Using Profiles](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [AssociateProfile AWS CLI Command Reference](#).

## associate-resource-to-profile

Il seguente esempio di codice mostra come utilizzare `associate-resource-to-profile`.

### AWS CLI

Associare una risorsa a un profilo

L'esempio seguente associa un gruppo di regole DNS Firewall con priorità 102 a un profilo.

```

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}'

```

Output:

```

{
  "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"
  }
}
```

- Per API i dettagli, vedere [AssociateResourceToProfile](#) in AWS CLI Command Reference.

## create-profile

Il seguente esempio di codice mostra come utilizzare `create-profile`.

### AWS CLI

Per creare un profilo

L'`create-profile` esempio seguente crea un profilo.

```
aws route53profiles create-profile \
  --name test
```

Output:

```
{
  "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"
  }
}
```



- Per API i dettagli, vedere [CreateProfile](#) in AWS CLI Command Reference.

## delete-profile

Il seguente esempio di codice mostra come utilizzare `delete-profile`.

### AWS CLI

Per eliminare un profilo

L'`delete-profile` esempio seguente elimina un profilo.

```
aws route53profiles delete-profile \  
  --profile-id rp-6ffe47d5example
```

Output:

```
{  
  "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"  
  }  
}
```

- Per API i dettagli, vedere [DeleteProfile](#) in AWS CLI Command Reference.

## disassociate-profile

Il seguente esempio di codice mostra come utilizzare `disassociate-profile`.

### AWS CLI

Per dissociare un profilo

L'`disassociate-profile` seguente dissocia un profilo da un VPC

```
aws route53profiles disassociate-profile \
  --profile-id rp-4987774726example \
  --resource-id vpc-0af3b96b3example
```

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [DisassociateProfile](#) in AWS CLI Command Reference.

## `disassociate-resource-from-profile`

Il seguente esempio di codice mostra come utilizzare `disassociate-resource-from-profile`.

AWS CLI

Per dissociare una risorsa da Profile

L'`disassociate-resource-from-profile` seguente dissocia un gruppo di regole DNS Firewall da un Profile.

```
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
```

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [DisassociateResourceFromProfile](#) in AWS CLI Command Reference.

## get-profile-association

Il seguente esempio di codice mostra come utilizzare `get-profile-association`.

### AWS CLI

Per ottenere informazioni su un'associazione di profili

Quanto segue `get-profile-association` restituisce informazioni sull'associazione di profili specificata.

```
aws route53profiles get-profile-association \
  --profile-association-id rpassoc-489ce212fexample
```

Output:

```
{
  "ProfileAssociation": {
    "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"
  }
}

```

- Per API i dettagli, vedere [GetProfileAssociation](#) in AWS CLI Command Reference.

## get-profile-resource-association

Il seguente esempio di codice mostra come utilizzare `get-profile-resource-association`.

### AWS CLI

Per ottenere informazioni su una risorsa associata a un profilo

quanto segue `get-profile-resource-association` restituisce informazioni sull'associazione di risorse specificata a un profilo.

```

aws route53profiles get-profile-resource-association \
  --profile-resource-association-id rpr-001913120a7example

```

Output:

```

{
  "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"
  }
}

```

```
}  
}
```

- Per API i dettagli, vedere [GetProfileResourceAssociation](#) in AWS CLI Command Reference.

## get-profile

Il seguente esempio di codice mostra come utilizzare `get-profile`.

### AWS CLI

Per ottenere informazioni su un profilo

Quanto segue `get-profile` restituisce informazioni sul profilo specificato.

```
aws route53profiles get-profile \  
  --profile-id rp-4987774726example
```

Output:

```
{  
  "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"  
  }  
}
```

- Per API i dettagli, vedere [GetProfile](#) in AWS CLI Command Reference.

## list-profile-associations

Il seguente esempio di codice mostra come utilizzare `list-profile-associations`.

## AWS CLI

Per elencare le associazioni dei profili

Di seguito `list-profile-associations` sono elencate le associazioni di profili presenti nel tuo AWS account.

```
aws route53profiles list-profile-associations
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ListProfileAssociations](#) in AWS CLI Command Reference.

## `list-profile-resource-associations`

Il seguente esempio di codice mostra come utilizzare `list-profile-resource-associations`.

## AWS CLI

Per elencare le associazioni di risorse del profilo

Di seguito sono `list-profile-resource-associations` elencate le associazioni di risorse Profile per il profilo specificato.

```
aws route53profiles list-profile-resource-associations \
```

```
--profile-id rp-4987774726example
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ListProfileResourceAssociations](#) in AWS CLI Command Reference.

## list-profiles

Il seguente esempio di codice mostra come utilizzare `list-profiles`.

AWS CLI

Per elencare i profili

Di seguito `list-profiles` sono elencati i profili presenti nell' AWS account e vengono visualizzate informazioni aggiuntive su di essi.

```
aws route53profiles list-profiles
```

Output:

```
{
```

```
"ProfileSummaries": [  
  {  
    "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/  
rp-4987774726example",  
    "Id": "rp-4987774726example",  
    "Name": "test",  
    "ShareStatus": "NOT_SHARED"  
  }  
]
```

- Per API i dettagli, vedere [ListProfiles](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una risorsa

Di seguito `list-tags-for-resource` sono elencati i tag per la risorsa specificata.

```
aws route53profiles list-tags-for-resource \  
  --resource-arn arn:aws:route53profiles:us-east-1:123456789012:profile/  
rp-4987774726example
```

Output:

```
{  
  "Tags": {  
    "my-key-2": "my-value-2",  
    "my-key-1": "my-value-1"  
  }  
}
```

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## update-profile-resource-association

Il seguente esempio di codice mostra come utilizzare `update-profile-resource-association`.



## AWS CLI

Per aggiornare una risorsa associata a un profilo

Quanto segue `update-profile-resource-association` aggiorna una priorità di un gruppo di regole del DNS firewall associato al profilo.

```
aws route53profiles update-profile-resource-association \
  --profile-resource-association-id rpr-001913120a7example \
  --resource-properties "{\"priority\": 105}"
```

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [UpdateProfileResourceAssociation](#) in AWS CLI Command Reference.

## Esempi di Route 53 Resolver che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Route 53 Resolver.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **associate-firewall-rule-group**

Il seguente esempio di codice mostra come utilizzare `associate-firewall-rule-group`.

#### AWS CLI

Per associare un gruppo di regole del firewall a VPC

L'esempio seguente associa un gruppo di regole DNS Firewall a un Amazon VPC.

```
aws route53resolver associate-firewall-rule-group \  
  --name test-association \  
  --firewall-rule-group-id rslvr-frg-47f93271fexample \  
  --vpc-id vpc-31e92222 \  
  --priority 101
```

#### Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consulta la sezione [Gestione delle associazioni tra il tuo gruppo di regole VPC e quello di Route 53 Resolver DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [AssociateFirewallRuleGroup](#) Reference.

## associate-resolver-endpoint-ip-address

Il seguente esempio di codice mostra come utilizzare `associate-resolver-endpoint-ip-address`.

### AWS CLI

Per associare un altro indirizzo IP a un endpoint Resolver

L'esempio seguente associa un altro indirizzo IP a un endpoint Resolver in ingresso. Se specificate solo un ID di sottorete e omettete l'indirizzo IP dal `--ip-address` parametro, Resolver sceglie automaticamente un indirizzo IP tra gli indirizzi IP disponibili nella sottorete specificata.

```

aws route53resolver associate-resolver-endpoint-ip-address \
  --resolver-endpoint-id rslvr-in-497098ad5example \
  --ip-address="SubnetId=subnet-12d8exam,Ip=192.0.2.118"

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Valori specificati durante la creazione o la modifica degli endpoint in entrata](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [AssociateResolverEndpointIpAddressReference](#).

## associate-resolver-rule

Il seguente esempio di codice mostra come utilizzare `associate-resolver-rule`.

### AWS CLI

Per associare una regola Resolver a VPC

L'esempio seguente associa una regola Resolver a un Amazon VPC. Dopo aver eseguito il comando, Resolver inizia a inoltrare DNS le query alla rete in base alle impostazioni della regola, ad esempio il nome di dominio delle query inoltrate.

```

aws route53resolver associate-resolver-rule \
  --name my-resolver-rule-association \
  --resolver-rule-id rslvr-rr-42b60677c0example \
  --vpc-id vpc-304bexam

```

Output:

```

{
  "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."
  }
}

```

```
}
```

Per ulteriori informazioni, consulta [Inoltro di DNS query in uscita alla rete](#) nella Amazon Route 53 Developer Guide.

- Per i API dettagli, consulta [AssociateResolverRule](#) Command Reference.AWS CLI

## create-firewall-domain-list

Il seguente esempio di codice mostra come utilizzare `create-firewall-domain-list`.

### AWS CLI

Per creare un elenco di domini Route 53 Resolver DNS Firewall

L'`create-firewall-domain-list`esempio seguente crea un elenco di domini Route 53 Resolver DNS Firewall, denominato `test`, nel tuo AWS account.

```
aws route53resolver create-firewall-domain-list \  
  --creator-request-id my-request-id \  
  --name test
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Managing your own domain lists](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [CreateFirewallDomainList AWS CLI](#) Command Reference.

## create-firewall-rule-group

Il seguente esempio di codice mostra come utilizzare `create-firewall-rule-group`.

### AWS CLI

Per creare un gruppo di regole del firewall

L'esempio seguente crea un gruppo di regole DNS Firewall.

```
aws route53resolver create-firewall-rule-group \  
  --creator-request-id my-request-id \  
  --name test
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Managing rule group and rules in DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [CreateFirewallRuleGroup AWS CLI Command Reference](#).

## create-firewall-rule

Il seguente esempio di codice mostra come utilizzare `create-firewall-rule`.

## AWS CLI

Per creare una regola firewall

L'create-firewall-rule esempio seguente crea una regola firewall in una regola DNS Firewall per i domini elencati in un elenco di domini DNS Firewall.

```
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
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Managing rule group and rules in DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [CreateFirewallRule AWS CLI](#) Command Reference.

## create-resolver-endpoint

Il seguente esempio di codice mostra come utilizzare create-resolver-endpoint.

## AWS CLI

Per creare un endpoint Resolver in entrata

L'`create-resolver-endpoint` seguente crea un endpoint Resolver in ingresso. È possibile utilizzare lo stesso comando per creare endpoint in entrata e in uscita.

```
aws route53resolver create-resolver-endpoint --name my-inbound-endpoint -- creator-request-id
2020-01-18:47 -- «sg-f62bexam» --direction --ip-addresses =subnet-ba47exam, Ip=192.0.2.255
=subnet-12d8exam, Ip=192.0.2.254 security-group-ids INBOUND SubnetId SubnetId
```

Output:

```
{
  "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"
  }
}
```

Per creare un endpoint Resolver in uscita

L'`create-resolver-endpoint` seguente crea un endpoint resolver in uscita utilizzando i valori del documento `-formatted.json`. JSON `create-outbound-resolver-endpoint.json`

```
aws route53resolver create-resolver-endpoint \
  --cli-input-json file://c:\temp\create-outbound-resolver-endpoint.json
```

Contenuto di `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"
  }
]
}

```

Per ulteriori informazioni, consulta [Resolving DNS Queries Between VPCs and Your Network](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateResolverEndpoint](#)Reference.

## create-resolver-rule

Il seguente esempio di codice mostra come utilizzare `create-resolver-rule`.

### AWS CLI

Per creare una regola Resolver

L'`create-resolver-rule`esempio seguente crea una regola di inoltro Resolver. La regola utilizza l'endpoint in uscita `rsldr-out-d 5e5920e37example` per inoltrare le query agli indirizzi IP `10.24.8.75` e `10.24.8.156`. DNS `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"

```

Output:

```

{
  "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 Per i [CreateResolverRule AWS CLI](#) dettagli, vedere in Command Reference.

## delete-firewall-domain-list

Il seguente esempio di codice mostra come utilizzare `delete-firewall-domain-list`.

## AWS CLI

Per eliminare un elenco di domini Route 53 Resolver DNS Firewall

L'`delete-firewall-domain-list` seguente elimina un elenco di domini Route 53 Resolver DNS Firewall, denominato `test`, nell'account. AWS

```
aws route53resolver delete-firewall-domain-list \
  --firewall-domain-list-id rslvr-fdl-9e956e9ffexample
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Managing your own domain lists](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [DeleteFirewallDomainList AWS CLI Command Reference](#).

## `delete-firewall-rule-group`

Il seguente esempio di codice mostra come utilizzare `delete-firewall-rule-group`.

## AWS CLI

Per eliminare un gruppo di regole del firewall

L'`delete-firewall-rule-group` seguente elimina un gruppo di regole del firewall.

```
aws route53resolver delete-firewall-rule-group \  
--firewall-rule-group-id rslvr-frg-47f93271fexample
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, consulta [Managing rule group and rules in DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [DeleteFirewallRuleGroup AWS CLI Command Reference](#).

## delete-firewall-rule

Il seguente esempio di codice mostra come utilizzare `delete-firewall-rule`.

### AWS CLI

Per eliminare una regola del firewall

L'`delete-firewall-rule` esempio seguente elimina una regola firewall specificata.

```
aws route53resolver delete-firewall-rule \  
--firewall-rule-group-id rslvr-frg-47f93271fexample \  
--firewall-domain-list-id rslvr-fdl-9e956e9ffexample
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Managing rule group and rules in DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [DeleteFirewallRule AWS CLI Command Reference](#).

## delete-resolver-endpoint

Il seguente esempio di codice mostra come utilizzare `delete-resolver-endpoint`.

### AWS CLI

Per eliminare un endpoint Resolver

L'`delete-resolver-endpoint`esempio seguente elimina l'endpoint specificato.

Importante Se si elimina un endpoint in entrata, DNS le query dalla rete non vengono più inoltrate a Resolver nella modalità specificata nell'endpoint. VPC Se si elimina un endpoint in uscita, Resolver interrompe l'inoltro delle DNS query dalla rete alla rete per le regole che specificano l'endpoint in uscita eliminato. VPC

```
aws route53resolver delete-resolver-endpoint \
  --resolver-endpoint-id rslvr-in-497098ad59example
```

Output:

```
{
  "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 Per [DeleteResolverEndpoint](#) AWS CLI dettagli, consulta Command Reference.

## delete-resolver-rule

Il seguente esempio di codice mostra come utilizzare `delete-resolver-rule`.

### AWS CLI

Per eliminare una regola Resolver

L'`delete-resolver-rule` esempio seguente elimina la regola specificata.

Nota Se una regola è associata a una regola VPCs, è necessario prima dissociare la regola dalla regola VPCs prima di poterla eliminare.

```

aws route53resolver delete-resolver-rule \
  --resolver-rule-id rslvr-rr-5b3809426bexample

```

Output:

```

{
  "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"
  }
}

```

- Per API i dettagli, vedere [DeleteResolverRule](#) in AWS CLI Command Reference.

## disassociate-firewall-rule-group

Il seguente esempio di codice mostra come utilizzare `disassociate-firewall-rule-group`.

### AWS CLI

Per dissociare un gruppo di regole del firewall da un VPC

L'esempio seguente dissocia un gruppo di regole DNS Firewall da un Amazon VPC

```

aws route53resolver disassociate-firewall-rule-group \
  --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta la sezione [Gestione delle associazioni tra il tuo gruppo di regole VPC e quello di Route 53 Resolver DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DisassociateFirewallRuleGroup](#)Reference.

## disassociate-resolver-endpoint-ip-address

Il seguente esempio di codice mostra come utilizzare `disassociate-resolver-endpoint-ip-address`.

### AWS CLI

Per dissociare un indirizzo IP da un endpoint Resolver

L'`disassociate-resolver-endpoint-ip-address`esempio seguente rimuove un indirizzo IP da un endpoint Resolver in entrata o in uscita specificato.

Nota Un endpoint deve avere almeno due indirizzi IP. Se un endpoint ha attualmente solo due indirizzi IP e desideri sostituire un indirizzo con un altro indirizzo, devi prima utilizzare [associate-resolver-endpoint-ip-address per associare il nuovo indirizzo](#) IP. Quindi puoi dissociare uno degli indirizzi IP originali dall'endpoint.

```

aws route53resolver disassociate-resolver-endpoint-ip-address \
  --resolver-endpoint-id rslvr-in-f9ab8a03f1example \
  --ip-address="SubnetId=subnet-12d8a459,Ip=172.31.40.121"

```

Output:



```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [DisassociateResolverEndpointIpAddress](#) in AWS CLI Command Reference.

## disassociate-resolver-rule

Il seguente esempio di codice mostra come utilizzare `disassociate-resolver-rule`.

### AWS CLI

Per dissociare una regola Resolver da un Amazon VPC

L'`disassociate-resolver-rule` esempio seguente rimuove l'associazione tra la regola Resolver specificata e quella specificata. VPC È possibile dissociare una regola da una VPC nelle seguenti circostanze:

Per DNS le query che hanno origine in questa regola VPC, si desidera che Resolver interrompa l'inoltro delle query alla rete per il nome di dominio specificato nella regola. Si desidera eliminare la regola di inoltro. Se una regola è attualmente associata a una o più regole VPCs, è necessario dissociarla da tutte prima di poterla eliminare. VPCs

```
aws route53resolver disassociate-resolver-rule \
```

```
--resolver-rule-id rslvr-rr-4955cb98ceexample \  
--vpc-id vpc-304bexam
```

Output:

```
{  
  "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"  
  }  
}
```

- Per API i dettagli, vedere [DisassociateResolverRule](#) in AWS CLI Command Reference.

## get-firewall-config

Il seguente esempio di codice mostra come utilizzare `get-firewall-config`.

### AWS CLI

Per ottenere una configurazione del firewall per un VPC

L'`get-firewall-config` seguente recupera il comportamento del DNS firewall per il valore specificato. VPC

```
aws route53resolver get-firewall-config \  
--resource-id vpc-31e9222
```

Output:

```
{  
  "FirewallConfig": {  
    "Id": "rslvr-fc-86016850cexample",  
    "ResourceId": "vpc-31e9222",  
    "OwnerId": "123456789012",  
    "FirewallFailOpen": "DISABLED"  
  }  
}
```

```
}
```

Per ulteriori informazioni, consulta la [VPCconfigurazione DNS del firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [GetFirewallConfig AWS CLI](#) Command Reference.

## get-firewall-domain-list

Il seguente esempio di codice mostra come utilizzare `get-firewall-domain-list`.

### AWS CLI

Per ottenere un elenco di domini Route 53 Resolver DNS Firewall

L'`get-firewall-domain-list` esempio seguente recupera l'elenco dei domini con l'ID specificato.

```
aws route53resolver get-firewall-domain-list \  
  --firewall-domain-list-id rs1vr-fdl-42b60677cexample
```

Output:

```
{  
  "FirewallDomainList": {  
    "Id": "rs1vr-fdl-9e956e9ffexample",  
    "Arn": "arn:aws:route53resolver:us-west-2:123457689012:firewall-domain-list/  
rs1vr-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"  
  }  
}
```

Per ulteriori informazioni, consulta [Managing your own domain lists](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [GetFirewallDomainList AWS CLI](#) Command Reference.

## get-firewall-rule-group-association

Il seguente esempio di codice mostra come utilizzare `get-firewall-rule-group-association`.

### AWS CLI

Per ottenere un'associazione tra gruppi di regole del firewall

L'`get-firewall-rule-group-association` seguente recupera un'associazione di gruppi di regole del firewall.

```
aws route53resolver get-firewall-rule-group-association \
  --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta la sezione [Gestione delle associazioni tra il tuo gruppo di regole VPC e quello di Route 53 Resolver DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetFirewallRuleGroupAssociation](#) Reference.

## get-firewall-rule-group-policy

Il seguente esempio di codice mostra come utilizzare `get-firewall-rule-group-policy`.

## AWS CLI

Per ottenere una AWS IAM politica

L'`get-firewall-rule-group-policy` seguente ottiene la politica AWS Identity and Access Management (AWS IAM) per la condivisione del gruppo di regole specificato.

```
aws route53resolver get-firewall-rule-group-policy \  
  --arn arn:aws:route53resolver:us-west-2:AWS_ACCOUNT_ID:firewall-rule-group/  
rslvr-frg-47f93271fexample
```

Output:

```
{  
  "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\"}]}"  
}
```

Per ulteriori informazioni, consulta [Managing rule group and rules in DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [GetFirewallRuleGroupPolicy AWS CLI Command Reference](#).

## `get-firewall-rule-group`

Il seguente esempio di codice mostra come utilizzare `get-firewall-rule-group`.

### AWS CLI

Per ottenere un gruppo di regole del firewall

L'`get-firewall-rule-group` seguente recupera informazioni su un gruppo di regole DNS Firewall con l'ID fornito.

```
aws route53resolver get-firewall-rule-group \  
  --firewall-rule-group-id rslvr-frg-47f93271fexample
```

**Output:**

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Managing rule group and rules in DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [GetFirewallRuleGroup AWS CLI Command Reference](#).

**get-resolver-endpoint**

Il seguente esempio di codice mostra come utilizzare `get-resolver-endpoint`.

**AWS CLI**

Per ottenere informazioni su un endpoint Resolver

L'`get-resolver-endpoint` seguente visualizza i dettagli per l'endpoint specificato in uscita. È possibile utilizzarlo sia `get-resolver-endpoint` per gli endpoint in entrata che per quelli in uscita specificando l'ID dell'endpoint applicabile.

```
aws route53resolver get-resolver-endpoint \
  --resolver-endpoint-id rslvr-out-d5e5920e37example
```

**Output:**

```
{
```

```

"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"
}
}

```

Per ulteriori informazioni, consulta [Valori specificati durante la creazione o la modifica degli endpoint in entrata](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetResolverEndpoint](#) Reference.

## get-resolver-rule-association

Il seguente esempio di codice mostra come utilizzare `get-resolver-rule-association`.

### AWS CLI

Per ottenere informazioni sull'associazione tra una regola Resolver e un VPC

L'esempio seguente visualizza i dettagli sull'associazione tra una regola Resolver specificata e una VPC. Si associa una regola del resolver e un using VPC [associate-resolver-rule](#)

```

aws route53resolver get-resolver-rule-association \
  --resolver-rule-association-id rslvr-rrassoc-d61cbb2c8bexample

```

Output:

```
{
```

```
"ResolverRuleAssociation": {
  "Id": "rslvr-rrassoc-d61cbb2c8bexample",
  "ResolverRuleId": "rslvr-rr-42b60677c0example",
  "Name": "my-resolver-rule-association",
  "VPCId": "vpc-304bexam",
  "Status": "COMPLETE",
  "StatusMessage": ""
}
}
```

- Per API i dettagli, vedere [GetResolverRuleAssociation](#) in AWS CLI Command Reference.

## get-resolver-rule

Il seguente esempio di codice mostra come utilizzare `get-resolver-rule`.

### AWS CLI

Per ottenere informazioni su una regola Resolver

L'`get-resolver-rule` esempio seguente visualizza i dettagli sulla regola Resolver specificata, ad esempio il nome di dominio per cui la regola inoltra le DNS query e l'ID dell'endpoint del resolver in uscita a cui è associata la regola.

```
aws route53resolver get-resolver-rule \
  --resolver-rule-id rslvr-rr-42b60677c0example
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Valori specificati durante la creazione o la modifica delle regole](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [GetResolverRule AWS CLI Command Reference](#).

## import-firewall-domains

Il seguente esempio di codice mostra come utilizzare `import-firewall-domains`.

### AWS CLI

Per importare domini in un elenco di domini

L'`import-firewall-domains` esempio seguente importa un set di domini da un file in un elenco di domini DNS Firewall specificato dall'utente.

```

aws route53resolver import-firewall-domains \
  --firewall-domain-list-id rslvr-fdl-d61cbb2cbexample \
  --operation REPLACE \
  --domain-file-url s3://PATH/TO/YOUR/FILE

```

Output:

```

{
  "Id": "rslvr-fdl-d61cbb2cbexample",
  "Name": "test",
  "Status": "IMPORTING",
  "StatusMessage": "Importing domains from provided file."
}

```

Per ulteriori informazioni, consulta [Managing your own domain lists](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [ImportFirewallDomains AWS CLI](#) Command Reference.

## **list-firewall-configs**

Il seguente esempio di codice mostra come utilizzare `list-firewall-configs`.

### AWS CLI

Per elencare le configurazioni del firewall

L'`list-firewall-configs` esempio seguente elenca le configurazioni DNS del firewall.

```
aws route53resolver list-firewall-configs
```

Output:

```
{
  "FirewallConfigs": [
    {
      "Id": "rslvr-fc-86016850cexample",
      "ResourceId": "vpc-31e92222",
      "OwnerId": "123456789012",
      "FirewallFailOpen": "DISABLED"
    }
  ]
}
```

Per ulteriori informazioni, consulta la [VPCconfigurazione DNS del firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [ListFirewallConfigs AWS CLI](#) Command Reference.

## **list-firewall-domain-lists**

Il seguente esempio di codice mostra come utilizzare `list-firewall-domain-lists`.

### AWS CLI

Per elencare tutti gli elenchi di domini di Route 53 Resolver DNS Firewall

L'`list-firewall-domain-lists` seguente elenca tutti gli elenchi di domini.

```
aws route53resolver list-firewall-domain-lists
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta gli [elenchi di domini Route 53 Resolver DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListFirewallDomainLists](#)Reference.

## `list-firewall-domains`

Il seguente esempio di codice mostra come utilizzare `list-firewall-domains`.

## AWS CLI

Per elencare i domini in un elenco di domini

L'`list-firewall-domains` seguente elenca i domini in un elenco di domini DNS Firewall specificati dall'utente.

```
aws route53resolver list-firewall-domains \
  --firewall-domain-list-id rs1vr-fdl-d61cbb2cbexample
```

Output:

```
{
  "Domains": [
    "test1.com.",
    "test2.com.",
    "test3.com."
  ]
}
```

Per ulteriori informazioni, consulta [Managing your own domain lists](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [ListFirewallDomains AWS CLI Command Reference](#).

## `list-firewall-rule-group-associations`

Il seguente esempio di codice mostra come utilizzare `list-firewall-rule-group-associations`.

## AWS CLI

Per elencare le associazioni dei gruppi di regole del DNS firewall

L'`list-firewall-rule-group-associations` seguente elenca le associazioni dei gruppi di regole del DNS firewall con AmazonVPCs.

```
aws route53resolver list-firewall-rule-group-associations
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Gestire le associazioni tra il tuo gruppo di regole VPC e quello di Route 53 Resolver DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListFirewallRuleGroupAssociations](#) Reference.

## list-firewall-rule-groups

Il seguente esempio di codice mostra come utilizzare `list-firewall-rule-groups`.

### AWS CLI

Per ottenere un elenco dei gruppi di regole del firewall

L'`list-firewall-rule-groups` esempio seguente elenca i gruppi di regole DNS del firewall.

```
aws route53resolver list-firewall-rule-groups
```

Output:

```
{
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Managing rule group and rules in DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [ListFirewallRuleGroups AWS CLI Command Reference](#).

## list-firewall-rules

Il seguente esempio di codice mostra come utilizzare `list-firewall-rules`.

### AWS CLI

Per elencare le regole del firewall

L'`list-firewall-rules` esempio seguente elenca tutte le regole del DNS firewall all'interno di un gruppo di regole firewall.

```

aws route53resolver list-firewall-rules \
  --firewall-rule-group-id rslvr-frg-47f93271fexample

```

Output:

```

{
  "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"
    }
  ]
}

```

```

    }
  ]
}

```

Per ulteriori informazioni, consulta [Managing rule group and rules in DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [ListFirewallRules AWS CLI Command Reference](#).

## list-resolver-endpoint-ip-addresses

Il seguente esempio di codice mostra come utilizzare `list-resolver-endpoint-ip-addresses`.

### AWS CLI

Per elencare gli indirizzi IP per un dispositivo in entrata o in uscita specificato

L'`list-resolver-endpoint-ip-addresses` esempio seguente elenca le informazioni sugli indirizzi IP associati all'endpoint in entrata. `rslvr-in-f9ab8a03f1example` È inoltre possibile utilizzarlo `list-resolver-endpoint-ip-addresses` per gli endpoint in uscita specificando l'ID dell'endpoint applicabile.

```

aws route53resolver list-resolver-endpoint-ip-addresses \
  --resolver-endpoint-id rslvr-in-f9ab8a03f1example

```

Output:

```

{
  "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"
    }
]
}
```

Per ulteriori informazioni sui valori nell'output, consulta [Valori che specifichi quando crei o modifichi gli endpoint in entrata](#) e [Valori specificati quando crei o modifichi gli endpoint in uscita](#), entrambi nella Amazon Route 53 Developer Guide.

- Per i API dettagli, consulta [ListResolverEndpointIpAddresses](#) Command Reference.AWS CLI

## list-resolver-endpoints

Il seguente esempio di codice mostra come utilizzare `list-resolver-endpoints`.

AWS CLI

Per elencare gli endpoint Resolver in una regione AWS

L'`list-resolver-endpoints` esempio seguente elenca gli endpoint Resolver in entrata e in uscita presenti nell'account corrente.

```
aws route53resolver list-resolver-endpoints
```

Output:

```
{
  "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"
    }
]
}

```

- Per i API dettagli, vedere in Command Reference. [ListResolverEndpoints](#) AWS CLI

## list-resolver-rule-associations

Il seguente esempio di codice mostra come utilizzare `list-resolver-rule-associations`.

### AWS CLI

Per elencare le associazioni tra le regole del Resolver e VPCs

L'`list-resolver-rule-associations` esempio seguente elenca le associazioni tra le regole del resolver e quelle VPCs nell'account corrente. AWS

```
aws route53resolver list-resolver-rule-associations
```

**Output:**

```
{
  "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": ""
    }
  ]
}
```

Per ulteriori informazioni, consulta [How Route 53 Resolver inoltra DNS le query dalla tua rete VPCs alla tua rete](#) nella Amazon Route 53 Developer Guide.

- Per i API dettagli, consulta [ListResolverRuleAssociations](#) Command Reference.AWS CLI

**list-resolver-rules**

Il seguente esempio di codice mostra come utilizzare `list-resolver-rules`.

**AWS CLI**

Per elencare le regole del Resolver

L'`list-resolver-rules` esempio seguente elenca tutte le regole Resolver nell'account corrente. AWS

```
aws route53resolver list-resolver-rules
```

## Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [How Route 53 Resolver inoltra DNS le query dalla tua rete VPCs alla tua rete](#) nella Amazon Route 53 Developer Guide.

- Per i API dettagli, consulta [ListResolverRules](#) Command Reference.AWS CLI

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una risorsa Resolver

L'`list-tags-for-resource` esempio seguente elenca i tag assegnati alla regola Resolver specificata.

```
aws route53resolver list-tags-for-resource \
  --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/
  rslvr-rr-42b60677c0example"
```

Output:

```
{
  "Tags": [
    {
      "Key": "my-key-1",
      "Value": "my-value-1"
    },
    {
      "Key": "my-key-2",
      "Value": "my-value-2"
    }
  ]
}
```

Per informazioni sull'utilizzo dei tag per l'allocazione dei costi, consulta [Using Cost Allocation Tag](#) nella AWS Billing and Cost Management User Guide.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## put-firewall-rule-group-policy

Il seguente esempio di codice mostra come utilizzare `put-firewall-rule-group-policy`.

## AWS CLI

Allegare una AWS IAM politica per condividere una politica del gruppo di regole del firewall

L'`put-firewall-rule-group-policy` esempio seguente allega un criterio AWS Identity and Access Management (AWS IAM) per la condivisione del gruppo di regole.

```
aws route53resolver put-firewall-rule-group-policy \
  --firewall-rule-group-policy "{\"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\"}]}"
```

Output:

```
{
  "ReturnValue": true
}
```

Per ulteriori informazioni, consulta [Managing rule group and rules in DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [PutFirewallRuleGroupPolicy AWS CLI Command Reference](#).

## `put-resolver-rule-policy`

Il seguente esempio di codice mostra come utilizzare `put-resolver-rule-policy`.

## AWS CLI

Per condividere una regola Resolver con un altro account AWS

L'`put-resolver-rule-policy` esempio seguente specifica una regola Resolver che si desidera condividere con un altro AWS account, l'account con cui si desidera condividere la regola e le operazioni relative alle regole che si desidera che l'account sia in grado di eseguire sulle regole.

Nota È necessario eseguire questo comando utilizzando le credenziali dello stesso account che ha creato la regola.

```
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\" ] } ] }"
```

Output:

```
{
  "ReturnValue": true
}
```

Dopo l'esecuzione di `put-resolver-rule-policy`, è possibile eseguire i due comandi seguenti di Resource Access Manager (RAM). È necessario utilizzare l'account con cui si desidera condividere la regola:

`get-resource-share-invitations` restituisce il valore `resourceShareInvitationArn`. È necessario questo valore per accettare l'invito a utilizzare la regola condivisa. `accept-resource-share-invitation` accetta l'invito a utilizzare la regola condivisa.

Per ulteriori informazioni, consulta la seguente documentazione :

[get-resource-share-invitationsaccept-resource-share-invitations](#) Condivisione delle regole di inoltro con altri AWS account e utilizzo di regole condivise nella Amazon Route 53 Developer Guide

- Per API i dettagli, consulta AWS CLI Command [PutResolverRulePolicy](#) Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

## AWS CLI

Per associare tag a una risorsa Resolver

L'`tag-resource` seguente associa due coppie chiave/valore di tag alla regola Resolver specificata.

```
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"
```

Questo comando non produce alcun output.

Per informazioni sull'utilizzo dei tag per l'allocazione dei costi, consulta [Using Cost Allocation Tag](#) nella AWS Billing and Cost Management User Guide.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

## AWS CLI

Per rimuovere i tag da una risorsa Resolver

L'`untag-resource` seguente rimuove due tag dalla regola Resolver specificata.

```
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
```

Questo comando non produce alcun output. Per confermare che i tag sono stati rimossi, puoi usare [list-tags-for-resource](#)

Per informazioni sull'utilizzo dei tag per l'allocazione dei costi, consulta [Using Cost Allocation Tag](#) nella AWS Billing and Cost Management User Guide.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-firewall-config

Il seguente esempio di codice mostra come utilizzare `update-firewall-config`.

### AWS CLI

Per aggiornare una configurazione del firewall

L'esempio seguente aggiorna la configurazione DNS del firewall.

```
aws route53resolver update-firewall-config \  
  --resource-id vpc-31e92222 \  
  --firewall-fail-open DISABLED
```

Output:

```
{  
  "FirewallConfig": {  
    "Id": "rslvr-fc-86016850cexample",  
    "ResourceId": "vpc-31e92222",  
    "OwnerId": "123456789012",  
    "FirewallFailOpen": "DISABLED"  
  }  
}
```

Per ulteriori informazioni, consulta la [VPCconfigurazione DNS del firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [UpdateFirewallConfig AWS CLI Command Reference](#).

## update-firewall-domains

Il seguente esempio di codice mostra come utilizzare `update-firewall-domains`.

### AWS CLI

Per aggiornare un elenco di domini

L'esempio seguente aggiunge i domini a un elenco di domini con l'ID fornito.

```
aws route53resolver update-firewall-domains \  
  --resource-id vpc-31e92222 \  
  --domains example.com
```



```
--firewall-domain-list-id rslvr-fdl-42b60677cexample \  
--operation ADD \  
--domains test1.com test2.com test3.com
```

Output:

```
{  
  "Id": "rslvr-fdl-42b60677cexample",  
  "Name": "test",  
  "Status": "UPDATING",  
  "StatusMessage": "Updating the Firewall Domain List"  
}
```

Per ulteriori informazioni, consulta [Managing your own domain lists](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [UpdateFirewallDomains AWS CLI Command Reference](#).

## update-firewall-rule-group-association

Il seguente esempio di codice mostra come utilizzare `update-firewall-rule-group-association`.

AWS CLI

Per aggiornare un'associazione di gruppi di regole del firewall

L'`update-firewall-rule-group-association`esempio seguente aggiorna un'associazione di gruppi di regole del firewall.

```
aws route53resolver update-firewall-rule-group-association \  
--firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example \  
--priority 103
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Gestire le associazioni tra il tuo gruppo di regole VPC e quello di Route 53 Resolver DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateFirewallRuleGroupAssociationReference](#).

## update-firewall-rule

Il seguente esempio di codice mostra come utilizzare `update-firewall-rule`.

### AWS CLI

Per aggiornare una regola del firewall

L'`update-firewall-rule` seguente aggiorna una regola firewall con i parametri specificati.

```

aws route53resolver update-firewall-rule \
  --firewall-rule-group-id rslvr-frg-47f93271fexample \
  --firewall-domain-list-id rslvr-fdl-9e956e9ffexample \
  --priority 102

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Managing rule group and rules in DNS Firewall](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [UpdateFirewallRule AWS CLI Command Reference](#).

## update-resolver-endpoint

Il seguente esempio di codice mostra come utilizzare `update-resolver-endpoint`.

### AWS CLI

Per aggiornare il nome di un endpoint Resolver

L'`update-resolver-endpoint` seguente aggiorna il nome di un endpoint Resolver. L'aggiornamento di altri valori non è supportato.

```

aws route53resolver update-resolver-endpoint \
  --resolver-endpoint-id rslvr-in-b5d45e32bdc445f09 \
  --name my-renamed-inbound-endpoint

```

Output:

```

{
  "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"
  }
}

```

- Per API i dettagli, vedere [UpdateResolverEndpoint](#) in AWS CLI Command Reference.

## update-resolver-rule

Il seguente esempio di codice mostra come utilizzare `update-resolver-rule`.

### AWS CLI

Esempio 1: Per aggiornare le impostazioni Resolver Endpoint

L'`update-resolver-rule` esempio seguente aggiorna il nome della regola, gli indirizzi IP sulla rete locale a cui vengono inoltrate DNS le query e l'ID dell'endpoint Resolver in uscita che stai utilizzando per inoltrare le query alla tua rete.

Nota I valori esistenti di `TargetIps` vengono sovrascritti, quindi è necessario specificare tutti gli indirizzi IP che si desidera assegnare alla regola dopo l'aggiornamento.

```

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

```

Output:

```

{
  "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"
  }
}

```

Esempio 2: Per aggiornare le impostazioni Resolver Endpoint utilizza un file per le impostazioni `config`

In alternativa, puoi includere le config impostazioni in un JSON file e poi specificare quel file quando chiami. `update-resolver-rule`

```

aws route53resolver update-resolver-rule \
  --resolver-rule-id rslvr-rr-1247fa64f3example \
  --config file://c:\temp\update-resolver-rule.json

```

Contenuto di `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"  
}
```

Per ulteriori informazioni, consulta [Valori specificati durante la creazione o la modifica delle regole](#) nella Amazon Route 53 Developer Guide.

- Per API i dettagli, consulta [UpdateResolverRule AWS CLI](#) Command Reference.

## Esempi di utilizzo di Amazon S3 AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon S3. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **abort-multipart-upload**

Il seguente esempio di codice mostra come utilizzare `abort-multipart-upload`.

AWS CLI

Per interrompere il caricamento multiparte specificato

Il `abort-multipart-upload` comando seguente interrompe un caricamento in più parti per la chiave `multipart/01` nel bucket `my-bucket`

```
aws s3api abort-multipart-upload \  
  --bucket my-bucket \  
  --key multipart/01 \  
  --
```

**--upload-**

```
id dFRtDYU0WMCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZLjF.Yxwh6XG7WfS2vC4to6HiV6YjLx.cph0gtNBtJ8P3UR
```

L'ID di caricamento richiesto da questo comando viene emesso da `create-multipart-upload` e può essere recuperato anche con `list-multipart-uploads`

- Per API i dettagli, vedere [AbortMultipartUpload](#) in AWS CLI Command Reference.

**complete-multipart-upload**

Il seguente esempio di codice mostra come utilizzare `complete-multipart-upload`.

## AWS CLI

Il comando seguente completa un caricamento in più parti per la chiave `multipart/01` nel bucket: `my-bucket`

```
aws s3api complete-multipart-upload --multipart-upload file://
mpustruct --bucket my-bucket --key 'multipart/01' --upload-
id dFRtDYU0WMCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZLjF.Yxwh6XG7WfS2vC4to6HiV6YjLx.cph0gtNBtJ8P3UR
```

L'ID di caricamento richiesto da questo comando viene emesso da `create-multipart-upload` e può essere recuperato anche con `list-multipart-uploads`

L'opzione di caricamento in più parti del comando precedente utilizza una JSON struttura che descrive le parti del caricamento in più parti che devono essere riassemblate nel file completo. In questo esempio, il `file://` prefisso viene utilizzato per caricare la JSON struttura da un file nella cartella locale denominata `mpustruct`

`mpustruct`:

```
{
  "Parts": [
    {
      "ETag": "e868e0f4719e394144ef36531ee6824c",
      "PartNumber": 1
    },
    {
      "ETag": "6bb2b12753d66fe86da4998aa33fffb0",
      "PartNumber": 2
    },
    {
```

```

    "ETag": "d0a0112e841abec9c9ec83406f0159c8",
    "PartNumber": 3
  }
]
}

```

Il ETag valore per ogni parte è upload, viene emesso ogni volta che si carica una parte utilizzando il `upload-part` comando e può anche essere recuperato chiamando `list-parts` o calcolato prendendo il MD5 checksum di ogni parte.

Output:

```

{
  "ETag": "\"3944a9f7a4faab7f78788ff6210f63f0-3\"",
  "Bucket": "my-bucket",
  "Location": "https://my-bucket.s3.amazonaws.com/multipart%2F01",
  "Key": "multipart/01"
}

```

- Per API i dettagli, vedere [CompleteMultipartUpload](#) in AWS CLI Command Reference.

## copy-object

Il seguente esempio di codice mostra come utilizzare `copy-object`.

AWS CLI

Il comando seguente copia un oggetto da `bucket-1` a `bucket-2`:

```

aws s3api copy-object --copy-source bucket-1/test.txt --key test.txt --
bucket bucket-2

```

Output:

```

{
  "CopyObjectResult": {
    "LastModified": "2015-11-10T01:07:25.000Z",
    "ETag": "\"589c8b79c230a6ecd5a7e1d040a9a030\""
  },
  "VersionId": "YdnYvTCVDqRRFA.NFJjy36p0hxifM1kA"
}

```



- Per API i dettagli, vedere [CopyObject](#) in AWS CLI Command Reference.

## cp

Il seguente esempio di codice mostra come utilizzare `cp`.

### AWS CLI

Esempio 1: copia di un file locale su S3

Il `cp` comando seguente copia un singolo file in un bucket e una chiave specificati:

```
aws s3 cp test.txt s3://mybucket/test2.txt
```

Output:

```
upload: test.txt to s3://mybucket/test2.txt
```

Esempio 2: copia di un file locale su S3 con una data di scadenza

Il `cp` comando seguente copia un singolo file in un bucket e in una chiave specificati che scadono al timestamp 8601 specificato: ISO

```
aws s3 cp test.txt s3://mybucket/test2.txt \  
--expires 2014-10-01T20:30:00Z
```

Output:

```
upload: test.txt to s3://mybucket/test2.txt
```

Esempio 3: copia di un file da S3 a S3

Il `cp` comando seguente copia un singolo oggetto s3 in un bucket e una chiave specificati:

```
aws s3 cp s3://mybucket/test.txt s3://mybucket/test2.txt
```

Output:

```
copy: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

#### Esempio 4: copia di un oggetto S3 in un file locale

Il cp comando seguente copia localmente un singolo oggetto in un file specificato:

```
aws s3 cp s3://mybucket/test.txt test2.txt
```

Output:

```
download: s3://mybucket/test.txt to test2.txt
```

#### Esempio 5: copia di un oggetto S3 da un bucket a un altro

Il cp comando seguente copia un singolo oggetto in un bucket specificato mantenendo il nome originale:

```
aws s3 cp s3://mybucket/test.txt s3://mybucket2/
```

Output:

```
copy: s3://mybucket/test.txt to s3://mybucket2/test.txt
```

#### Esempio 6: copia ricorsiva di oggetti S3 in una directory locale

Quando viene passato con il parametro `--recursive`, il cp comando seguente copia in modo ricorsivo tutti gli oggetti con un prefisso e un bucket specificati in una directory specificata. In questo esempio, il bucket contiene gli oggetti e: mybucket test1.txt test2.txt

```
aws s3 cp s3://mybucket . \  
--recursive
```

Output:

```
download: s3://mybucket/test1.txt to test1.txt  
download: s3://mybucket/test2.txt to test2.txt
```

#### Esempio 7: copia ricorsiva di file locali su S3

Quando viene passato con il parametro `--recursive`, il cp comando seguente copia in modo ricorsivo tutti i file in una directory specificata in un bucket e un prefisso specificati, escludendo

alcuni file utilizzando un parametro. `--exclude` In questo esempio, la directory `myDir` contiene i file `e: test1.txt test2.jpg`

```
aws s3 cp myDir s3://mybucket/ \  
  --recursive \  
  --exclude "*.jpg"
```

Output:

```
upload: myDir/test1.txt to s3://mybucket/test1.txt
```

Esempio 8: copia ricorsiva di oggetti S3 in un altro bucket

Quando viene passato con il parametro `--recursive`, il `cp` comando seguente copia in modo ricorsivo tutti gli oggetti contenuti in un bucket specificato in un altro bucket, escludendo alcuni oggetti utilizzando un parametro. `--exclude` In questo esempio, il bucket contiene gli oggetti `e: mybucket test1.txt another/test1.txt`

```
aws s3 cp s3://mybucket/ s3://mybucket2/ \  
  --recursive \  
  --exclude "another/*"
```

Output:

```
copy: s3://mybucket/test1.txt to s3://mybucket2/test1.txt
```

È possibile combinare `--exclude` `--include` diverse opzioni per copiare solo gli oggetti che corrispondono a uno schema, escludendo tutti gli altri:

```
aws s3 cp s3://mybucket/logs/ s3://mybucket2/logs/ \  
  --recursive \  
  --exclude "*" \  
  --include "*.log"
```

Output:

```
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
```

## Esempio 9: impostazione dell'Access Control List (ACL) durante la copia di un oggetto S3

Il cp comando seguente copia un singolo oggetto in un bucket e in una chiave specificati mentre lo imposta su: ACL public-read-write

```
aws s3 cp s3://mybucket/test.txt s3://mybucket/test2.txt \  
--acl public-read-write
```

Output:

```
copy: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

Tieni presente che se utilizzi l'--acl opzione, assicurati che tutte IAM le politiche associate includano l'"s3:PutObjectAcl" azione:

```
aws iam get-user-policy \  
--user-name myuser \  
--policy-name mypolicy
```

Output:

```
{  
  "UserName": "myuser",  
  "PolicyName": "mypolicy",  
  "PolicyDocument": {  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Action": [  
          "s3:PutObject",  
          "s3:PutObjectAcl"  
        ],  
        "Resource": [  
          "arn:aws:s3:::mybucket/*"  
        ],  
        "Effect": "Allow",  
        "Sid": "Stmt1234567891234"  
      }  
    ]  
  }  
}
```

### Esempio 10: concessione delle autorizzazioni per un oggetto S3

Il cp comando seguente illustra l'uso dell'- --grantsoptione per concedere l'accesso in lettura a tutti gli utenti identificati URI e il pieno controllo a un utente specifico identificato dal relativo ID canonico:

```
aws s3 cp file.txt s3://mybucket/ --grants read=uri=http://acs.amazonaws.com/groups/global/AllUsers full=id=79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Output:

```
upload: file.txt to s3://mybucket/file.txt
```

### Esempio 11: caricamento di un flusso di file locale su S3

PowerShell può alterare la codifica o aggiungere un input CRLF a piped.

Il cp comando seguente carica un flusso di file locale dallo standard input a un bucket e una chiave specificati:

```
aws s3 cp - s3://mybucket/stream.txt
```

### Esempio 12: caricamento di un flusso di file locale di dimensioni superiori a 50 GB su S3

Il cp comando seguente carica un flusso di file locale da 51 GB dall'input standard a un bucket e una chiave specificati. È necessario fornire l'- --expected-sizeopzione, altrimenti il caricamento potrebbe fallire quando raggiunge il limite di 10.000 parti predefinito:

```
aws s3 cp - s3://mybucket/stream.txt --expected-size 54760833024
```

### Esempio 13: scaricare un oggetto S3 come flusso di file locale

PowerShell può alterare la codifica o aggiungere un output in piping o CRLF reindirizzato.

Il cp comando seguente scarica un oggetto S3 localmente come stream verso lo standard output. Il download come stream non è attualmente compatibile con il --recursive parametro:

```
aws s3 cp s3://mybucket/stream.txt -
```

### Esempio 14: caricamento su un punto di accesso S3

Il cp comando seguente carica un singolo file (mydoc.txt) nel punto di accesso () in corrispondenza della chiave (myaccesspoint): mykey

```
aws s3 cp mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/  
myaccesspoint/mykey
```

Output:

```
upload: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/  
myaccesspoint/mykey
```

### Esempio 15: download da un punto di accesso S3

Il cp comando seguente scarica un singolo oggetto (mykey) dal punto di accesso (myaccesspoint) al file locale (mydoc.txt):

```
aws s3 cp s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/  
mykey mydoc.txt
```

Output:

```
download: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey to  
mydoc.txt
```

- Per API i dettagli, vedete [Cp](#) in AWS CLI Command Reference.

## create-bucket

Il seguente esempio di codice mostra come usare create-bucket.

### AWS CLI

Esempio 1: creare un bucket

L'create-bucketesempio seguente crea un bucket denominato: my-bucket

```
aws s3api create-bucket \
```

```
--bucket my-bucket \  
--region us-east-1
```

Output:

```
{  
  "Location": "/my-bucket"  
}
```

Per ulteriori informazioni, consulta [Creare un bucket nella Guida](#) per l'utente di Amazon S3.

Esempio 2: creare un bucket con il proprietario imposto

L'`create-bucket` seguente crea un bucket denominato `my-bucket` che utilizza l'impostazione `bucket owner enforced` per S3 Object Ownership.

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region us-east-1 \  
  --object-ownership BucketOwnerEnforced
```

Output:

```
{  
  "Location": "/my-bucket"  
}
```

Per ulteriori informazioni, consulta la sezione [Controllo della proprietà degli oggetti e disattivazione ACLs nella Guida](#) per l'utente di Amazon S3.

Esempio 3: creare un bucket al di fuori della regione `us-east-1`

L'`create-bucket` seguente crea un bucket denominato `my-bucket` nella regione `eu-west-1`. Le aree esterne `LocationConstraint` a `us-east-1` richiedono che venga specificato il valore appropriato per creare il bucket nella regione desiderata.

```
aws s3api create-bucket \  
  --bucket my-bucket \  
  --region eu-west-1 \  
  --create-bucket-configuration LocationConstraint=eu-west-1
```

Output:

```
{
  "Location": "http://my-bucket.s3.amazonaws.com/"
}
```

Per ulteriori informazioni, consulta [Creare un bucket nella Guida](#) per l'utente di Amazon S3.

- Per API i dettagli, consulta AWS CLI Command [CreateBucket](#)Reference.

## create-multipart-upload

Il seguente esempio di codice mostra come utilizzare `create-multipart-upload`.

AWS CLI

Il comando seguente crea un caricamento in più parti nel bucket `my-bucket` con la chiave: `multipart/01`

```
aws s3api create-multipart-upload --bucket my-bucket --key 'multipart/01'
```

Output:

```
{
  "Bucket": "my-bucket",
  "UploadId":
  "dfRtDYU0WWCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZ1jF.Yxwh6XG7WfS2vC4to6HiV6Yj1x.cph0gtNBtJ8P3URC
  "Key": "multipart/01"
}
```

Il file completato verrà denominato `01` in una cartella chiamata `multipart` nel bucket. `my-bucket` Salva l'ID di caricamento, la chiave e il nome del bucket da utilizzare con il `upload-part` comando.

- Per API i dettagli, vedere [CreateMultipartUpload](#)in AWS CLI Command Reference.

## delete-bucket-analytics-configuration

Il seguente esempio di codice mostra come utilizzare `delete-bucket-analytics-configuration`.



## AWS CLI

Per eliminare una configurazione di analisi per un bucket

L'`delete-bucket-analytics-configuration` seguente rimuove la configurazione di analisi per il bucket e l'ID specificati.

```
aws s3api delete-bucket-analytics-configuration \  
  --bucket my-bucket \  
  --id 1
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteBucketAnalyticsConfiguration](#) in AWS CLI Command Reference.

## `delete-bucket-cors`

Il seguente esempio di codice mostra come utilizzare `delete-bucket-cors`.

### AWS CLI

Il comando seguente elimina una configurazione Cross-Origin Resource Sharing da un bucket denominato: `my-bucket`

```
aws s3api delete-bucket-cors --bucket my-bucket
```

- Per API i dettagli, vedere [DeleteBucketCors](#) in AWS CLI Command Reference.

## `delete-bucket-encryption`

Il seguente esempio di codice mostra come utilizzare `delete-bucket-encryption`.

### AWS CLI

Per eliminare la configurazione di crittografia lato server di un bucket

L'`delete-bucket-encryption` seguente elimina la configurazione di crittografia lato server del bucket specificato.

```
aws s3api delete-bucket-encryption \  
  --bucket my-bucket
```

```
--bucket my-bucket
```

Questo comando non produce alcun output.

- Per i API dettagli, vedere [DeleteBucketEncryption](#) in Command Reference.AWS CLI

## **delete-bucket-intelligent-tiering-configuration**

Il seguente esempio di codice mostra come utilizzare `delete-bucket-intelligent-tiering-configuration`.

### AWS CLI

Per rimuovere una configurazione S3 Intelligent-Tiering su un bucket

L'`delete-bucket-intelligent-tiering-configuration` esempio seguente rimuove una configurazione S3 Intelligent-Tiering, denominata, su un bucket. `ExampleConfig`

```
aws s3api delete-bucket-intelligent-tiering-configuration \  
  --bucket DOC-EXAMPLE-BUCKET \  
  --id ExampleConfig
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Using S3 Intelligent-Tiering nella](#) Amazon S3 User Guide.

- Per i API dettagli, consulta Command Reference.  
[DeleteBucketIntelligentTieringConfiguration](#) AWS CLI

## **delete-bucket-inventory-configuration**

Il seguente esempio di codice mostra come utilizzare `delete-bucket-inventory-configuration`.

### AWS CLI

Per eliminare la configurazione di inventario di un bucket

L'`delete-bucket-inventory-configuration` esempio seguente elimina la configurazione dell'inventario con ID 1 per il bucket specificato.

```
aws s3api delete-bucket-inventory-configuration \  
  --bucket my-bucket \  
  --id 1
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteBucketInventoryConfiguration](#) in AWS CLI Command Reference.

## delete-bucket-lifecycle

Il seguente esempio di codice mostra come utilizzare `delete-bucket-lifecycle`.

AWS CLI

Il comando seguente elimina una configurazione del ciclo di vita da un bucket denominato: `my-bucket`

```
aws s3api delete-bucket-lifecycle --bucket my-bucket
```

- Per i API dettagli, vedere [DeleteBucketLifecycle](#) in Command Reference.AWS CLI

## delete-bucket-metrics-configuration

Il seguente esempio di codice mostra come utilizzare `delete-bucket-metrics-configuration`.

AWS CLI

Per eliminare una configurazione di metriche per un bucket

L'`delete-bucket-metrics-configuration` esempio seguente rimuove la configurazione delle metriche per il bucket e l'ID specificati.

```
aws s3api delete-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123
```

Questo comando non produce alcun output.

- Per API i dettagli, vedete [DeleteBucketMetricsConfiguration](#) in AWS CLI Command Reference.

## delete-bucket-ownership-controls

Il seguente esempio di codice mostra come utilizzare `delete-bucket-ownership-controls`.

### AWS CLI

Per rimuovere le impostazioni di proprietà del bucket di un bucket

L'esempio seguente rimuove le impostazioni di proprietà del bucket di un bucket.

```
aws s3api delete-bucket-ownership-controls \
  --bucket DOC-EXAMPLE-BUCKET
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Setting Object Ownership su un bucket esistente](#) nella Amazon S3 User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteBucketOwnershipControls](#) Reference.

## delete-bucket-policy

Il seguente esempio di codice mostra come utilizzare `delete-bucket-policy`.

### AWS CLI

Il comando seguente elimina una policy relativa ai bucket da un bucket denominato: `my-bucket`

```
aws s3api delete-bucket-policy --bucket my-bucket
```

- Per API i dettagli, vedere [DeleteBucketPolicy](#) in AWS CLI Command Reference.

## delete-bucket-replication

Il seguente esempio di codice mostra come utilizzare `delete-bucket-replication`.

### AWS CLI

Il comando seguente elimina una configurazione di replica da un bucket denominato: `my-bucket`

```
aws s3api delete-bucket-replication --bucket my-bucket
```

- Per API i dettagli, vedere [DeleteBucketReplication](#) in AWS CLI Command Reference.

## delete-bucket-tagging

Il seguente esempio di codice mostra come utilizzare `delete-bucket-tagging`.

### AWS CLI

Il comando seguente elimina una configurazione di tagging da un bucket denominato: `my-bucket`

```
aws s3api delete-bucket-tagging --bucket my-bucket
```

- Per API i dettagli, vedere [DeleteBucketTagging](#) in AWS CLI Command Reference.

## delete-bucket-website

Il seguente esempio di codice mostra come utilizzare `delete-bucket-website`.

### AWS CLI

Il comando seguente elimina la configurazione di un sito Web da un bucket denominato: `my-bucket`

```
aws s3api delete-bucket-website --bucket my-bucket
```

- Per API i dettagli, vedere [DeleteBucketWebsite](#) in AWS CLI Command Reference.

## delete-bucket

Il seguente esempio di codice mostra come utilizzare `delete-bucket`.

### AWS CLI

Il comando seguente elimina un bucket denominato: `my-bucket`

```
aws s3api delete-bucket --bucket my-bucket --region us-east-1
```

- Per API i dettagli, vedere [DeleteBucket](#) in AWS CLI Command Reference.

## delete-object-tagging

Il seguente esempio di codice mostra come utilizzare `delete-object-tagging`.

### AWS CLI

Per eliminare i set di tag di un oggetto

L'esempio seguente elimina il tag con la chiave specificata dall'oggetto `doc1.rtf`.

```
aws s3api delete-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteObjectTagging](#) in AWS CLI Command Reference.

## delete-object

Il seguente esempio di codice mostra come utilizzare `delete-object`.

### AWS CLI

Il comando seguente elimina un oggetto denominato `test.txt` da un bucket denominato: `my-bucket`

```
aws s3api delete-object --bucket my-bucket --key test.txt
```

Se il controllo delle versioni del bucket è abilitato, l'output conterrà l'ID di versione del marker di eliminazione:

```
{  
  "VersionId": "9_gKg5vG56F.TTEUdwkxGpJ3tND1w1Gq",  
  "DeleteMarker": true  
}
```

Per ulteriori informazioni sull'eliminazione di oggetti, consulta [Deleting Objects](#) nella [Amazon S3 Developer Guide](#).

- Per API i dettagli, consulta Command [DeleteObject](#)Reference AWS CLI .

## delete-objects

Il seguente esempio di codice mostra come utilizzare `delete-objects`.

### AWS CLI

Il comando seguente elimina un oggetto da un bucket denominato: `my-bucket`

```
aws s3api delete-objects --bucket my-bucket --delete file://delete.json
```

`delete.json` è un JSON documento nella directory corrente che specifica l'oggetto da eliminare:

```
{
  "Objects": [
    {
      "Key": "test1.txt"
    }
  ],
  "Quiet": false
}
```

Output:

```
{
  "Deleted": [
    {
      "DeleteMarkerVersionId": "mYAT5Mc6F7aeUL8SS7FAAqUP01koHwzU",
      "Key": "test1.txt",
      "DeleteMarker": true
    }
  ]
}
```

- Per API i dettagli, vedere [DeleteObjects](#)in AWS CLI Command Reference.

## delete-public-access-block

Il seguente esempio di codice mostra come utilizzare `delete-public-access-block`.

## AWS CLI

Per eliminare la configurazione di blocco dell'accesso pubblico per un bucket

L'`delete-public-access-block` seguente rimuove la configurazione di accesso pubblico a blocchi sul bucket specificato.

```
aws s3api delete-public-access-block \  
  --bucket my-bucket
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeletePublicAccessBlock](#) in AWS CLI Command Reference.

## `get-bucket-accelerate-configuration`

Il seguente esempio di codice mostra come utilizzare `get-bucket-accelerate-configuration`.

## AWS CLI

Per recuperare la configurazione di accelerazione di un bucket

L'`get-bucket-accelerate-configuration` seguente recupera la configurazione di accelerazione per il bucket specificato.

```
aws s3api get-bucket-accelerate-configuration \  
  --bucket my-bucket
```

Output:

```
{  
  "Status": "Enabled"  
}
```

- Per API i dettagli, vedere [GetBucketAccelerateConfiguration](#) in AWS CLI Command Reference.

## `get-bucket-acl`

Il seguente esempio di codice mostra come utilizzare `get-bucket-acl`.



## AWS CLI

Il comando seguente recupera l'elenco di controllo degli accessi per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-acl --bucket my-bucket
```

Output:

```
{
  "Owner": {
    "DisplayName": "my-username",
    "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
      },
      "Permission": "FULL_CONTROL"
    }
  ]
}
```

- Per API i dettagli, vedere [GetBucketAcl](#) in AWS CLI Command Reference.

## get-bucket-analytics-configuration

Il seguente esempio di codice mostra come utilizzare `get-bucket-analytics-configuration`.

### AWS CLI

Per recuperare la configurazione di analisi per un bucket con un ID specifico

L'`get-bucket-analytics-configuration` esempio seguente mostra la configurazione di analisi per il bucket e l'ID specificati.

```
aws s3api get-bucket-analytics-configuration \
  --bucket my-bucket \
```

```
--id 1
```

Output:

```
{
  "AnalyticsConfiguration": {
    "StorageClassAnalysis": {},
    "Id": "1"
  }
}
```

- Per API i dettagli, vedere [GetBucketAnalyticsConfiguration](#) in AWS CLI Command Reference.

## get-bucket-cors

Il seguente esempio di codice mostra come utilizzare `get-bucket-cors`.

AWS CLI

Il comando seguente recupera la configurazione Cross-Origin Resource Sharing per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-cors --bucket my-bucket
```

Output:

```
{
  "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": [
        "*"
      ]
    }
  ]
}

```

- Per API i dettagli, vedere [GetBucketCors](#) in AWS CLI Command Reference.

## get-bucket-encryption

Il seguente esempio di codice mostra come utilizzare `get-bucket-encryption`.

### AWS CLI

Per recuperare la configurazione di crittografia lato server per un bucket

L'`get-bucket-encryption` esempio seguente recupera la configurazione di crittografia lato server per il bucket `my-bucket`

```
aws s3api get-bucket-encryption \
  --bucket my-bucket
```

Output:

```

{
  "ServerSideEncryptionConfiguration": {
    "Rules": [
      {
        "ApplyServerSideEncryptionByDefault": {

```

```

    "SSEAlgorithm": "AES256"
  }
}
]
}
}

```

- Per i API dettagli, vedere [GetBucketEncryption](#) in Command Reference.AWS CLI

## get-bucket-intelligent-tiering-configuration

Il seguente esempio di codice mostra come utilizzare `get-bucket-intelligent-tiering-configuration`.

### AWS CLI

Per recuperare una configurazione S3 Intelligent-Tiering su un bucket

L'`get-bucket-intelligent-tiering-configuration` esempio seguente recupera una configurazione S3 Intelligent-Tiering, denominata, su un bucket. `ExampleConfig`

```

aws s3api get-bucket-intelligent-tiering-configuration \
  --bucket DOC-EXAMPLE-BUCKET \
  --id ExampleConfig

```

Output:

```

{
  "IntelligentTieringConfiguration": {
    "Id": "ExampleConfig2",
    "Filter": {
      "Prefix": "images"
    },
    "Status": "Enabled",
    "Tierings": [
      {
        "Days": 90,
        "AccessTier": "ARCHIVE_ACCESS"
      },
      {
        "Days": 180,
        "AccessTier": "DEEP_ARCHIVE_ACCESS"
      }
    ]
  }
}

```

```

    }
  ]
}
}

```

Per ulteriori informazioni, consulta [Using S3 Intelligent-Tiering nella Amazon S3 User Guide](#).

- Per i API dettagli, consulta Command Reference. [GetBucketIntelligentTieringConfiguration](#) AWS CLI

## get-bucket-inventory-configuration

Il seguente esempio di codice mostra come utilizzare `get-bucket-inventory-configuration`.

AWS CLI

Per recuperare la configurazione di inventario per un bucket

L'`get-bucket-inventory-configuration` esempio seguente recupera la configurazione dell'inventario per il bucket specificato con ID. 1

```

aws s3api get-bucket-inventory-configuration \
  --bucket my-bucket \
  --id 1

```

Output:

```

{
  "InventoryConfiguration": {
    "IsEnabled": true,
    "Destination": {
      "S3BucketDestination": {
        "Format": "ORC",
        "Bucket": "arn:aws:s3:::my-bucket",
        "AccountId": "123456789012"
      }
    },
    "IncludedObjectVersions": "Current",
    "Id": "1",
    "Schedule": {
      "Frequency": "Weekly"
    }
  }
}

```

```
}  
}
```

- Per API i dettagli, vedere [GetBucketInventoryConfiguration](#) in AWS CLI Command Reference.

## get-bucket-lifecycle-configuration

Il seguente esempio di codice mostra come utilizzare `get-bucket-lifecycle-configuration`.

### AWS CLI

Il comando seguente recupera la configurazione del ciclo di vita per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-lifecycle-configuration --bucket my-bucket
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

```
]
}
```

- Per i API dettagli, vedere [GetBucketLifecycleConfiguration](#) in Command Reference.AWS CLI

## get-bucket-lifecycle

Il seguente esempio di codice mostra come utilizzare `get-bucket-lifecycle`.

### AWS CLI

Il comando seguente recupera la configurazione del ciclo di vita per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-lifecycle --bucket my-bucket
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per i API dettagli, vedere [GetBucketLifecycle](#) in Command Reference.AWS CLI

## get-bucket-location

Il seguente esempio di codice mostra come utilizzare `get-bucket-location`.

### AWS CLI

Il comando seguente recupera il vincolo di posizione per un bucket denominato `my-bucket`, se esiste un vincolo:

```
aws s3api get-bucket-location --bucket my-bucket
```

Output:

```
{
  "LocationConstraint": "us-west-2"
}
```

- Per i API dettagli, vedere in Command Reference. [GetBucketLocation](#) AWS CLI

## get-bucket-logging

Il seguente esempio di codice mostra come utilizzare `get-bucket-logging`.

### AWS CLI

Per recuperare lo stato di registrazione di un bucket

L'`get-bucket-logging` seguente recupera lo stato di registrazione per il bucket specificato.

```
aws s3api get-bucket-logging \
  --bucket my-bucket
```

Output:

```
{
  "LoggingEnabled": {
    "TargetPrefix": "",
    "TargetBucket": "my-bucket-logs"
  }
}
```



- Per API i dettagli, vedere [GetBucketLogging](#) in AWS CLI Command Reference.

## get-bucket-metrics-configuration

Il seguente esempio di codice mostra come utilizzare `get-bucket-metrics-configuration`.

### AWS CLI

Per recuperare la configurazione delle metriche per un bucket con un ID specifico

L'esempio seguente visualizza la configurazione delle metriche per il bucket e l'ID specificati.

```
aws s3api get-bucket-metrics-configuration \
  --bucket my-bucket \
  --id 123
```

Output:

```
{
  "MetricsConfiguration": {
    "Filter": {
      "Prefix": "logs"
    },
    "Id": "123"
  }
}
```

- Per API i dettagli, vedete [GetBucketMetricsConfiguration](#) in AWS CLI Command Reference.

## get-bucket-notification-configuration

Il seguente esempio di codice mostra come utilizzare `get-bucket-notification-configuration`.

### AWS CLI

Il comando seguente recupera la configurazione di notifica per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-notification-configuration --bucket my-bucket
```

**Output:**

```
{
  "TopicConfigurations": [
    {
      "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNI",
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
      "Events": [
        "s3:ObjectCreated:*"
      ]
    }
  ]
}
```

- Per API i dettagli, vedere [GetBucketNotificationConfiguration](#) in AWS CLI Command Reference.

**get-bucket-notification**

Il seguente esempio di codice mostra come utilizzare `get-bucket-notification`.

**AWS CLI**

Il comando seguente recupera la configurazione di notifica per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-notification --bucket my-bucket
```

**Output:**

```
{
  "TopicConfiguration": {
    "Topic": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
    "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNI",
    "Event": "s3:ObjectCreated:*",
    "Events": [
      "s3:ObjectCreated:*"
    ]
  }
}
```

- Per API i dettagli, vedere [GetBucketNotification](#) in AWS CLI Command Reference.

## get-bucket-ownership-controls

Il seguente esempio di codice mostra come utilizzare `get-bucket-ownership-controls`.

### AWS CLI

Per recuperare le impostazioni di proprietà del bucket di un bucket

L'`get-bucket-ownership-controls` esempio seguente recupera le impostazioni di proprietà del bucket di un bucket.

```
aws s3api get-bucket-ownership-controls \
  --bucket DOC-EXAMPLE-BUCKET
```

Output:

```
{
  "OwnershipControls": {
    "Rules": [
      {
        "ObjectOwnership": "BucketOwnerEnforced"
      }
    ]
  }
}
```

Per ulteriori informazioni, consulta [Visualizzazione dell'impostazione della proprietà dell'oggetto per un bucket S3](#) nella Amazon S3 User Guide.

- Per API i dettagli, consulta Command [GetBucketOwnershipControls](#) Reference AWS CLI .

## get-bucket-policy-status

Il seguente esempio di codice mostra come utilizzare `get-bucket-policy-status`.

### AWS CLI

Per recuperare lo stato della politica di un bucket che indica se il bucket è pubblico

L'`get-bucket-policy-status` esempio seguente recupera lo stato della politica per il bucket.  
`my-bucket`

```
aws s3api get-bucket-policy-status \
  --bucket my-bucket
```

Output:

```
{
  "PolicyStatus": {
    "IsPublic": false
  }
}
```

- Per API i dettagli, vedere [GetBucketPolicyStatus](#) in AWS CLI Command Reference.

## get-bucket-policy

Il seguente esempio di codice mostra come utilizzare `get-bucket-policy`.

AWS CLI

Il comando seguente recupera la policy del bucket per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-policy --bucket my-bucket
```

Output:

```
{
  "Policy": "{\n\"Version\": \"2008-10-17\", \"Statement\": [\n{\n\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": \"*\", \"Action\": \"s3:GetObject\", \"Resource\": \"arn:aws:s3:::my-bucket/*\"}, {\n\"Sid\": \"\", \"Effect\": \"Deny\", \"Principal\": \"*\", \"Action\": \"s3:GetObject\", \"Resource\": \"arn:aws:s3:::my-bucket/secret/*\"}]]}"
}
```

Il policyThe seguente esempio mostra come scaricare una policy sui bucket di Amazon S3, apportare modifiche al file e quindi `put-bucket-policy` utilizzarla per applicare la policy del bucket modificata. Per scaricare la bucket policy in un file, puoi eseguire:

```
aws s3api get-bucket-policy --bucket mybucket --query Policy --output text >
policy.json
```

È quindi possibile modificare il `policy.json` file in base alle esigenze. Infine puoi riapplicare questa policy modificata al bucket S3 eseguendo:

`policy.json` file se necessario. Infine puoi riapplicare questa policy modificata al bucket S3 eseguendo:

file se necessario. Infine puoi riapplicare questa policy modificata al bucket S3 eseguendo:

```
aws s3api put-bucket-policy --bucket mybucket --policy file://policy.json
```

- Per API i dettagli, consulta AWS CLI Command [GetBucketPolicy](#) Reference.

## get-bucket-replication

Il seguente esempio di codice mostra come utilizzare `get-bucket-replication`.

### AWS CLI

Il comando seguente recupera la configurazione di replica per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-replication --bucket my-bucket
```

Output:

```
{
  "ReplicationConfiguration": {
    "Rules": [
      {
        "Status": "Enabled",
        "Prefix": "",
        "Destination": {
          "Bucket": "arn:aws:s3:::my-bucket-backup",
          "StorageClass": "STANDARD"
        },
        "ID": "ZmUwNzE4ZmQ4tMjVhOS00MT1kLOGI4NDkzZTIWJjNTUtYTA1"
      }
    ],
    "Role": "arn:aws:iam::123456789012:role/s3-replication-role"
  }
}
```

- Per API i dettagli, vedere [GetBucketReplication](#) in AWS CLI Command Reference.

## get-bucket-request-payment

Il seguente esempio di codice mostra come utilizzare `get-bucket-request-payment`.

### AWS CLI

Per recuperare la configurazione di pagamento della richiesta per un bucket

L'esempio seguente recupera la configurazione `requester pays` per il bucket specificato.

```
aws s3api get-bucket-request-payment \
  --bucket my-bucket
```

Output:

```
{
  "Payer": "BucketOwner"
}
```

- Per API i dettagli, vedere [GetBucketRequestPayment](#) in AWS CLI Command Reference.

## get-bucket-tagging

Il seguente esempio di codice mostra come utilizzare `get-bucket-tagging`.

### AWS CLI

Il comando seguente recupera la configurazione di tagging per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-tagging --bucket my-bucket
```

Output:

```
{
  "TagSet": [
    {
      "Value": "marketing",
      "Key": "organization"
    }
  ]
}
```

```
}
```

- Per API i dettagli, vedere [GetBucketTagging](#) in AWS CLI Command Reference.

## get-bucket-versioning

Il seguente esempio di codice mostra come utilizzare `get-bucket-versioning`.

### AWS CLI

Il comando seguente recupera la configurazione del controllo delle versioni per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-versioning --bucket my-bucket
```

Output:

```
{
  "Status": "Enabled"
}
```

- Per API i dettagli, vedere [GetBucketVersioning](#) in AWS CLI Command Reference.

## get-bucket-website

Il seguente esempio di codice mostra come utilizzare `get-bucket-website`.

### AWS CLI

Il comando seguente recupera la configurazione statica del sito Web per un bucket denominato: `my-bucket`

```
aws s3api get-bucket-website --bucket my-bucket
```

Output:

```
{
  "IndexDocument": {
    "Suffix": "index.html"
  },
}
```

```
"ErrorDocument": {
  "Key": "error.html"
}
}
```

- Per API i dettagli, vedere [GetBucketWebsite](#) in AWS CLI Command Reference.

## get-object-acl

Il seguente esempio di codice mostra come utilizzare `get-object-acl`.

### AWS CLI

Il comando seguente recupera l'elenco di controllo degli accessi per un oggetto in un bucket denominato: `my-bucket`

```
aws s3api get-object-acl --bucket my-bucket --key index.html
```

Output:

```
{
  "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"
    }
  ]
}
```



- Per API i dettagli, vedere [GetObjectACL](#) in AWS CLI Command Reference.

## get-object-attributes

Il seguente esempio di codice mostra come utilizzare `get-object-attributes`.

### AWS CLI

Recuperare i metadati da un oggetto senza restituire l'oggetto stesso

L'`get-object-attributes` esempio seguente recupera i metadati dall'oggetto. `doc1.rtf`

```
aws s3api get-object-attributes \  
  --bucket my-bucket \  
  --key doc1.rtf \  
  --object-attributes "StorageClass" "ETag" "ObjectSize"
```

Output:

```
{  
  "LastModified": "2022-03-15T19:37:31+00:00",  
  "VersionId": "IuCPjXTDzHNf1dAuitVBIKJpF2p1fg4P",  
  "ETag": "b662d79adeb7c8d787ea7eafb9ef6207",  
  "StorageClass": "STANDARD",  
  "ObjectSize": 405  
}
```

Per ulteriori informazioni, [GetObjectAttributes](#) consulta Amazon S3 API Reference.

- Per API i dettagli, consulta [GetObjectAttributes AWS CLI](#) Command Reference.

## get-object-legal-hold

Il seguente esempio di codice mostra come utilizzare `get-object-legal-hold`.

### AWS CLI

Recupera lo stato di conservazione legale di un oggetto

L'`get-object-legal-hold` esempio seguente recupera lo stato Legal Hold per l'oggetto specificato.

```
aws s3api get-object-legal-hold \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf
```

Output:

```
{  
  "LegalHold": {  
    "Status": "ON"  
  }  
}
```

- Per API i dettagli, vedere [GetObjectLegalHold](#) in AWS CLI Command Reference.

## get-object-lock-configuration

Il seguente esempio di codice mostra come utilizzare `get-object-lock-configuration`.

### AWS CLI

Per recuperare una configurazione di blocco degli oggetti per un bucket

L'`get-object-lock-configuration` esempio seguente recupera la configurazione del blocco degli oggetti per il bucket specificato.

```
aws s3api get-object-lock-configuration \  
  --bucket my-bucket-with-object-lock
```

Output:

```
{  
  "ObjectLockConfiguration": {  
    "ObjectLockEnabled": "Enabled",  
    "Rule": {  
      "DefaultRetention": {  
        "Mode": "COMPLIANCE",  
        "Days": 50  
      }  
    }  
  }  
}
```

- Per API i dettagli, vedere [GetObjectLockConfiguration](#) in AWS CLI Command Reference.

## get-object-retention

Il seguente esempio di codice mostra come utilizzare `get-object-retention`.

### AWS CLI

Per recuperare la configurazione di conservazione degli oggetti per un oggetto

L'`get-object-retention` seguente recupera la configurazione di conservazione degli oggetti per l'oggetto specificato.

```
aws s3api get-object-retention \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf
```

Output:

```
{  
  "Retention": {  
    "Mode": "GOVERNANCE",  
    "RetainUntilDate": "2025-01-01T00:00:00.000Z"  
  }  
}
```

- Per API i dettagli, vedere [GetObjectRetention](#) in AWS CLI Command Reference.

## get-object-tagging

Il seguente esempio di codice mostra come utilizzare `get-object-tagging`.

### AWS CLI

Per recuperare i tag allegati a un oggetto

L'`get-object-tagging` seguente recupera i valori per la chiave specificata dall'oggetto specificato.

```
aws s3api get-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf
```

```
--key doc1.rtf
```

Output:

```
{
  "TagSet": [
    {
      "Value": "confidential",
      "Key": "designation"
    }
  ]
}
```

L'get-object-taggingesempio seguente tenta di recuperare i set di tag dell'oggettodoc2.rtf, che non ha tag.

```
aws s3api get-object-tagging \
  --bucket my-bucket \
  --key doc2.rtf
```

Output:

```
{
  "TagSet": []
}
```

L'get-object-taggingesempio seguente recupera i set di tag dell'oggettodoc3.rtf, che ha più tag.

```
aws s3api get-object-tagging \
  --bucket my-bucket \
  --key doc3.rtf
```

Output:

```
{
  "TagSet": [
    {
      "Value": "confidential",
      "Key": "designation"
    }
  ]
}
```

```
    },  
    {  
      "Value": "finance",  
      "Key": "department"  
    },  
    {  
      "Value": "payroll",  
      "Key": "team"  
    }  
  ]  
}
```

- Per API i dettagli, vedete [GetObjectTagging](#) in AWS CLI Command Reference.

## get-object-torrent

Il seguente esempio di codice mostra come utilizzare `get-object-torrent`.

### AWS CLI

Il comando seguente crea un torrent per un oggetto in un bucket denominato: `my-bucket`

```
aws s3api get-object-torrent --bucket my-bucket --key large-video-file.mp4 large-video-file.torrent
```

Il file torrent viene salvato localmente nella cartella corrente. Notate che il nome del file di output (`large-video-file.torrent`) è specificato senza un nome di opzione e deve essere l'ultimo argomento del comando.

- Per API i dettagli, vedere [GetObjectTorrent](#) in AWS CLI Command Reference.

## get-object

Il seguente esempio di codice mostra come utilizzare `get-object`.

### AWS CLI

L'esempio seguente utilizza il `get-object` comando per scaricare un oggetto da Amazon S3:

```
aws s3api get-object --bucket text-content --key dir/my_images.tar.bz2 my_images.tar.bz2
```

Tieni presente che il parametro `outfile` è specificato senza un nome di opzione come `--outfile`. Il nome del file di output deve essere l'ultimo parametro del comando.

L'esempio seguente dimostra l'utilizzo di `--range` per scaricare un intervallo di byte specifico da un oggetto. Nota che gli intervalli di byte devono avere il prefisso `bytes=`:

```
aws s3api get-object --bucket text-content --key dir/my_data --  
range bytes=8888-9999 my_data_range
```

Per ulteriori informazioni sul recupero di oggetti, consulta [Getting Objects](#) nella *Amazon S3 Developer Guide*.

- Per API i dettagli, consulta [Command `GetObject` Reference AWS CLI](#).

## get-public-access-block

Il seguente esempio di codice mostra come utilizzare `get-public-access-block`.

### AWS CLI

Per impostare o modificare la configurazione di accesso pubblico a blocchi per un bucket

L'esempio seguente visualizza la configurazione dell'accesso pubblico a blocchi per il bucket specificato.

```
aws s3api get-public-access-block \  
--bucket my-bucket
```

Output:

```
{  
  "PublicAccessBlockConfiguration": {  
    "IgnorePublicAcls": true,  
    "BlockPublicPolicy": true,  
    "BlockPublicAcls": true,  
    "RestrictPublicBuckets": true  
  }  
}
```

- Per API i dettagli, vedere [GetPublicAccessBlock](#) in *AWS CLI Command Reference*.

## head-bucket

Il seguente esempio di codice mostra come utilizzare `head-bucket`.

### AWS CLI

Il comando seguente verifica l'accesso a un bucket denominato: `my-bucket`

```
aws s3api head-bucket --bucket my-bucket
```

Se il bucket esiste e si ha accesso ad esso, non viene restituito alcun output. In caso contrario, verrà visualizzato un messaggio di errore. Per esempio:

```
A client error (404) occurred when calling the HeadBucket operation: Not Found
```

- Per API i dettagli, vedere [HeadBucket](#) in AWS CLI Command Reference.

## head-object

Il seguente esempio di codice mostra come utilizzare `head-object`.

### AWS CLI

Il comando seguente recupera i metadati per un oggetto in un bucket denominato: `my-bucket`

```
aws s3api head-object --bucket my-bucket --key index.html
```

Output:

```
{
  "AcceptRanges": "bytes",
  "ContentType": "text/html",
  "LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
  "ContentLength": 77,
  "VersionId": "null",
  "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
  "Metadata": {}
}
```

- Per API i dettagli, vedete [HeadObject](#) in AWS CLI Command Reference.

## list-bucket-analytics-configurations

Il seguente esempio di codice mostra come utilizzare `list-bucket-analytics-configurations`.

### AWS CLI

Per recuperare un elenco di configurazioni di analisi per un bucket

Quanto segue `list-bucket-analytics-configurations` recupera un elenco di configurazioni di analisi per il bucket specificato.

```
aws s3api list-bucket-analytics-configurations \
  --bucket my-bucket
```

Output:

```
{
  "AnalyticsConfigurationList": [
    {
      "StorageClassAnalysis": {},
      "Id": "1"
    }
  ],
  "IsTruncated": false
}
```

- Per API i dettagli, vedere [ListBucketAnalyticsConfigurations](#) in AWS CLI Command Reference.

## list-bucket-intelligent-tiering-configurations

Il seguente esempio di codice mostra come utilizzare `list-bucket-intelligent-tiering-configurations`.

### AWS CLI

Per recuperare tutte le configurazioni S3 Intelligent-Tiering su un bucket

L'`list-bucket-intelligent-tiering-configurations`esempio seguente recupera tutte le configurazioni di S3 Intelligent-Tiering su un bucket.



```
aws s3api list-bucket-intelligent-tiering-configurations \  
--bucket DOC-EXAMPLE-BUCKET
```

Output:

```
{  
  "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"
      }
    ]
  }
]
```

Per ulteriori informazioni, consulta [Using S3 Intelligent-Tiering nella Amazon S3 User Guide](#).

- Per i API dettagli, consulta Command Reference.  
[ListBucketIntelligentTieringConfigurations](#) AWS CLI

## list-bucket-inventory-configurations

Il seguente esempio di codice mostra come utilizzare `list-bucket-inventory-configurations`.

AWS CLI

Per recuperare un elenco di configurazioni di inventario per un bucket

L'`list-bucket-inventory-configurations` esempio seguente elenca le configurazioni di inventario per il bucket specificato.

```
aws s3api list-bucket-inventory-configurations \
  --bucket my-bucket
```

Output:

```
{
  "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
}

```

- Per API i dettagli, vedere [ListBucketInventoryConfigurations](#) in AWS CLI Command Reference.

## list-bucket-metrics-configurations

Il seguente esempio di codice mostra come utilizzare `list-bucket-metrics-configurations`.

### AWS CLI

Per recuperare un elenco di configurazioni di metriche per un bucket

L'`list-bucket-metrics-configurations` esempio seguente recupera un elenco di configurazioni di metriche per il bucket specificato.

```
aws s3api list-bucket-metrics-configurations \
  --bucket my-bucket
```

Output:

```
{
  "IsTruncated": false,
  "MetricsConfigurationList": [
    {
      "Filter": {
        "Prefix": "logs"
      },
      "Id": "123"
    },
    {
      "Filter": {
        "Prefix": "tmp"
      },
      "Id": "234"
    }
  ]
}
```

- Per i API dettagli, vedete [ListBucketMetricsConfigurations](#) in Command Reference.AWS CLI

## list-buckets

Il seguente esempio di codice mostra come utilizzare `list-buckets`.

AWS CLI

Il comando seguente utilizza il `list-buckets` comando per visualizzare i nomi di tutti i bucket Amazon S3 (in tutte le regioni):

```
aws s3api list-buckets --query "Buckets[].Name"
```

L'opzione `query` filtra l'output dei `list-buckets` soli nomi dei bucket.

Per ulteriori informazioni sui bucket, consulta [Working with Amazon S3 Buckets](#) nella Amazon S3 Developer Guide.

- Per i API dettagli, consulta [ListBuckets](#) Command Reference.AWS CLI

## list-multipart-uploads

Il seguente esempio di codice mostra come utilizzare `list-multipart-uploads`.

### AWS CLI

Il comando seguente elenca tutti i caricamenti multiparte attivi per un bucket denominato: `my-bucket`

```
aws s3api list-multipart-uploads --bucket my-bucket
```

Output:

```
{
  "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": []
}
```

In corso, i caricamenti multiparte comportano costi di archiviazione in Amazon S3. Completa o annulla un caricamento multiparte attivo per rimuoverne le parti dal tuo account.

- Per API i dettagli, consulta AWS CLI Command [ListMultipartUploads](#) Reference.

## list-object-versions

Il seguente esempio di codice mostra come utilizzare `list-object-versions`.

### AWS CLI

Il comando seguente recupera le informazioni sulla versione di un oggetto in un bucket denominato: `my-bucket`

```
aws s3api list-object-versions --bucket my-bucket --prefix index.html
```

Output:

```
{
  "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
  }
]
}

```

- Per API i dettagli, vedere [ListObjectVersions](#) in AWS CLI Command Reference.

## list-objects-v2

Il seguente esempio di codice mostra come utilizzare `list-objects-v2`.

### AWS CLI

Per ottenere un elenco di oggetti in un bucket

L'`list-objects-v2` esempio seguente elenca gli oggetti nel bucket specificato.

```
aws s3api list-objects-v2 \  
  --bucket my-bucket
```

Output:

```
{  
  "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  
    }  
  ]  
}
```



```

    ]
  }
}

```

- Per API i dettagli, vedere [ListObjectsV2](#) in AWS CLI Command Reference.

## list-objects

Il seguente esempio di codice mostra come utilizzare `list-objects`.

### AWS CLI

L'esempio seguente utilizza il `list-objects` comando per visualizzare i nomi di tutti gli oggetti nel bucket specificato:

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size: Size}'
```

L'esempio utilizza l'`--query` argomento per filtrare l'output di `list-objects` fino al valore e alla dimensione della chiave per ogni oggetto

Per ulteriori informazioni sugli oggetti, consulta [Working with Amazon S3 Objects](#) nella Amazon S3 Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListObjects](#) Reference.

## list-parts

Il seguente esempio di codice mostra come utilizzare `list-parts`.

### AWS CLI

Il comando seguente elenca tutte le parti che sono state caricate per un caricamento in più parti con chiave `multipart/01` nel `my-bucket` bucket:

```
aws s3api list-parts --bucket my-bucket --key 'multipart/01' --upload-id dfRtDYU0WCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

Output:

```
{
```

```

"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"
}

```

- Per API i dettagli, vedere [ListParts](#) in AWS CLI Command Reference.

## ls

Il seguente esempio di codice mostra come utilizzare `ls`.

### AWS CLI

Esempio 1: elenco di tutti i bucket di proprietà dell'utente

Il `ls` comando seguente elenca tutti i bucket di proprietà dell'utente. In questo esempio, l'utente possiede i bucket `mybucket` e `mybucket2`. Il timestamp è la data di creazione del bucket,

mostrata nel fuso orario della macchina. Questa data può cambiare quando si apportano modifiche al bucket, ad esempio quando si modifica la politica relativa al bucket. Nota che se `s3://` viene utilizzato per l'argomento `path<S3Uri>`, elencherà anche tutti i bucket.

```
aws s3 ls
```

Output:

```
2013-07-11 17:08:50 mybucket
2013-07-24 14:55:44 mybucket2
```

Esempio 2: elenco di tutti i prefissi e gli oggetti in un bucket

Il `ls` comando seguente elenca gli oggetti e i prefissi comuni in un bucket e un prefisso specificati. In questo esempio, l'utente è proprietario del bucket con gli oggetti e. `mybucket test.txt somePrefix/test.txt` | `LastWriteTime` e `Length` sono arbitrari. Nota che, poiché il `ls` comando non ha alcuna interazione con il filesystem locale, `s3://` URI lo schema non è necessario per risolvere l'ambiguità e può essere omesso.

```
aws s3 ls s3://mybucket
```

Output:

```
                PRE somePrefix/
2013-07-25 17:06:27      88 test.txt
```

Esempio 3: elenco di tutti i prefissi e gli oggetti in un bucket e un prefisso specifici

Il `ls` comando seguente elenca gli oggetti e i prefissi comuni in un bucket e un prefisso specificati. Tuttavia, non ci sono oggetti né prefissi comuni nel bucket e nel prefisso specificati.

```
aws s3 ls s3://mybucket/noExistPrefix
```

Output:

```
None
```

Esempio 4: elenco ricorsivo di tutti i prefissi e gli oggetti in un bucket

Il `ls` comando seguente elencherà in modo ricorsivo gli oggetti in un bucket. Invece di essere visualizzati `PRE dirname/` nell'output, tutto il contenuto di un bucket verrà elencato in ordine.

```
aws s3 ls s3://mybucket \  
--recursive
```

Output:

```
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
```

Esempio 5: riepilogo di tutti i prefissi e gli oggetti in un bucket

Il `ls` comando seguente illustra lo stesso comando utilizzando le opzioni `--human-readable` e `--summarize`. `Bytes/MiB/KiB/GiB/TiB/PiB/EiB--human-readable` visualizza la dimensione del file in. `--summarize` mostra il numero totale di oggetti e la dimensione totale alla fine dell'elenco dei risultati:

```
aws s3 ls s3://mybucket \  
--recursive \  
--human-readable \  
--summarize
```

Output:

```
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
```

Esempio 6: elenco da un punto di accesso S3

Il seguente `ls` comando elenca gli oggetti dal punto di accesso (`myaccesspoint`):

```
aws s3 ls s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

Output:

```
                PRE somePrefix/
2013-07-25 17:06:27      88 test.txt
```

- Per API i dettagli, vedete [Ls](#) in AWS CLI Command Reference.

## mb

Il seguente esempio di codice mostra come usare `mb`.

### AWS CLI

Esempio 1: creare un bucket

Il `mb` comando seguente crea un bucket. In questo esempio, l'utente crea il `mybucket` bucket. Il bucket viene creato nella regione specificata nel file di configurazione dell'utente:

```
aws s3 mb s3://mybucket
```

Output:

```
make_bucket: s3://mybucket
```

Esempio 2: crea un bucket nella regione specificata

Il `mb` comando seguente crea un bucket in una regione specificata dal `--region` parametro. In questo esempio, l'utente crea il bucket `mybucket` nella regione: `us-west-1`

```
aws s3 mb s3://mybucket \  
  --region us-west-1
```

Output:

```
make_bucket: s3://mybucket
```

- Per API i dettagli, consulta [Mb](#) in AWS CLI Command Reference.

## mv

Il seguente esempio di codice mostra come utilizzare `mv`.

### AWS CLI

Esempio 1: sposta un file locale nel bucket specificato

Il `mv` comando seguente sposta un singolo file in un bucket e una chiave specificati.

```
aws s3 mv test.txt s3://mybucket/test2.txt
```

Output:

```
move: test.txt to s3://mybucket/test2.txt
```

Esempio 2: sposta un oggetto nel bucket e nella chiave specificati

Il `mv` comando seguente sposta un singolo oggetto S3 in un bucket e una chiave specificati.

```
aws s3 mv s3://mybucket/test.txt s3://mybucket/test2.txt
```

Output:

```
move: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

Esempio 3: sposta un oggetto S3 nella directory locale

Il `mv` comando seguente sposta localmente un singolo oggetto in un file specificato.

```
aws s3 mv s3://mybucket/test.txt test2.txt
```

**Output:**

```
move: s3://mybucket/test.txt to test2.txt
```

Esempio 4: sposta un oggetto con il suo nome originale nel bucket specificato

Il mv comando seguente sposta un singolo oggetto in un bucket specificato mantenendo il nome originale:

```
aws s3 mv s3://mybucket/test.txt s3://mybucket2/
```

**Output:**

```
move: s3://mybucket/test.txt to s3://mybucket2/test.txt
```

Esempio 5: sposta tutti gli oggetti e i prefissi di un bucket nella directory locale

Quando viene passato con il parametro `--recursive`, il mv comando seguente sposta in modo ricorsivo tutti gli oggetti con un prefisso e un bucket specificati in una directory specificata. In questo esempio, il bucket contiene gli oggetti `e.mybucket test1.txt test2.txt`

```
aws s3 mv s3://mybucket . \  
--recursive
```

**Output:**

```
move: s3://mybucket/test1.txt to test1.txt  
move: s3://mybucket/test2.txt to test2.txt
```

Esempio 6: Sposta tutti gli oggetti e i prefissi di un bucket nella directory locale, ad eccezione dei file ``.jpg``

Quando viene passato con il parametro `--recursive`, il mv comando seguente sposta in modo ricorsivo tutti i file in una directory specificata in un bucket e un prefisso specificati, escludendo alcuni file utilizzando un parametro. `--exclude` In questo esempio, la directory `myDir` contiene i file `e.test1.txt test2.jpg`

```
aws s3 mv myDir s3://mybucket/ \
```

```
--recursive \  
--exclude "*.jpg"
```

Output:

```
move: myDir/test1.txt to s3://mybucket2/test1.txt
```

Esempio 7: sposta tutti gli oggetti e i prefissi in un bucket nella directory locale, tranne il prefisso specificato

Quando viene passato con il parametro `--recursive`, il `mv` comando seguente sposta in modo ricorsivo tutti gli oggetti contenuti in un bucket specificato in un altro bucket, escludendo alcuni oggetti utilizzando un parametro. `--exclude` In questo esempio, il bucket contiene gli oggetti `mybucket test1.txt` e `another/test1.txt`

```
aws s3 mv s3://mybucket/ s3://mybucket2/ \  
--recursive \  
--exclude "mybucket/another/*"
```

Output:

```
move: s3://mybucket/test1.txt to s3://mybucket2/test1.txt
```

Esempio 8: sposta un oggetto nel bucket specificato e imposta il ACL

Il `mv` comando seguente sposta un singolo oggetto in un bucket e una chiave specificati durante l'ACL impostazione di `public-read-write`

```
aws s3 mv s3://mybucket/test.txt s3://mybucket/test2.txt \  
--acl public-read-write
```

Output:

```
move: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

Esempio 9: sposta un file locale nel bucket specificato e concedi le autorizzazioni

Il `mv` comando seguente illustra l'uso dell' `--grant` opzione per concedere l'accesso in lettura a tutti gli utenti e il controllo completo a un utente specifico identificato dal relativo indirizzo e-mail.



```
aws s3 mv file.txt s3://mybucket/ \  
--grants read=uri=http://acs.amazonaws.com/groups/global/  
AllUsers full=emailaddress=user@example.com
```

Output:

```
move: file.txt to s3://mybucket/file.txt
```

Esempio 10: sposta un file su un punto di accesso S3

Il mv comando seguente sposta un singolo file denominato `mydoc.txt` nel punto di accesso denominato `myaccesspoint` con la chiave denominata `mykey`.

```
aws s3 mv mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/  
myaccesspoint/mykey
```

Output:

```
move: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/  
mykey
```

- Per API i dettagli, vedere [Mv in AWS CLI Command Reference](#).

## presign

Il seguente esempio di codice mostra come usare `presign`.

### AWS CLI

Esempio 1: creare un oggetto prefirato URL con durata predefinita di un'ora che si collega a un oggetto in un bucket S3

Il `presign` comando seguente genera una chiave e una chiave prefirato URL per un bucket e una chiave specificati validi per un'ora.

```
aws s3 presign s3://DOC-EXAMPLE-BUCKET/test2.txt
```

Output:

```
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=EXAMBLE1234494d5fba3fed607f98018e1dfc62e2529ae96d844123456
```

Esempio 2: creare un oggetto prefirmato URL con una durata personalizzata che si collega a un oggetto in un bucket S3

Il `presign` comando seguente genera una chiave e una chiave prefirmati URL per un bucket e una chiave specificati validi per una settimana.

```
aws s3 presign s3://DOC-EXAMPLE-BUCKET/test2.txt \
  --expires-in 604800
```

Output:

```
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=EXAMBLE1234494d5fba3fed607f98018e1dfc62e2529ae96d844123456
```

Per ulteriori informazioni, consulta [Share an Object with Others](#) nella guida per sviluppatori S3.

- Per API i dettagli, consulta [Presign](#) in AWS CLI Command Reference.

## put-bucket-accelerate-configuration

Il seguente esempio di codice mostra come utilizzare `put-bucket-accelerate-configuration`.

AWS CLI

Per impostare la configurazione di accelerazione di un bucket

L'`put-bucket-accelerate-configuration` esempio seguente abilita la configurazione di accelerazione per il bucket specificato.

```
aws s3api put-bucket-accelerate-configuration \
  --bucket my-bucket \
  --accelerate-configuration Status=Enabled
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutBucketAccelerateConfiguration](#) in AWS CLI Command Reference.

## put-bucket-acl

Il seguente esempio di codice mostra come utilizzare `put-bucket-acl`.

### AWS CLI

Questo esempio concede `full control` a due AWS utenti (`user1@example.com` e `user2@example.com`) e `read` l'autorizzazione a tutti:

```
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
```

Vedi <http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html> per i dettagli sulla personalizzazione ACLs (i ACL comandi `s3api`, ad esempio `put-bucket-acl`, usano la stessa notazione abbreviata degli argomenti).

- Per i API dettagli, vedere in Command Reference. [PutBucketAcl](#) AWS CLI

## put-bucket-analytics-configuration

Il seguente esempio di codice mostra come utilizzare `put-bucket-analytics-configuration`.

### AWS CLI

Per impostare una configurazione di analisi per il bucket

L'`put-bucket-analytics-configuration` esempio seguente configura l'analisi per il bucket specificato.

```
aws s3api put-bucket-analytics-configuration \  
  --bucket my-bucket --id 1 \  
  --analytics-configuration '{"Id": "1", "StorageClassAnalysis": {}}'
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutBucketAnalyticsConfiguration](#) in AWS CLI Command Reference.

## put-bucket-cors

Il seguente esempio di codice mostra come utilizzare `put-bucket-cors`.

### AWS CLI

L'esempio seguente abilita PUT e DELETE richieste da `www.example.com` e abilita GET le richieste da qualsiasi dominio: POST

```
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
    }
  ]
}
```

- Per API i dettagli, vedere [PutBucketCors](#) in AWS CLI Command Reference.

## put-bucket-encryption

Il seguente esempio di codice mostra come utilizzare `put-bucket-encryption`.

### AWS CLI

Per configurare la crittografia lato server per un bucket

L'`put-bucket-encryption` esempio seguente imposta la AES256 crittografia come predefinita per il bucket specificato.

```
aws s3api put-bucket-encryption \  
  --bucket my-bucket \  
  --server-side-encryption-configuration '{"Rules":  
  [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}'
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutBucketEncryption](#) in AWS CLI Command Reference.

## put-bucket-intelligent-tiering-configuration

Il seguente esempio di codice mostra come utilizzare `put-bucket-intelligent-tiering-configuration`.

### AWS CLI

Per aggiornare una configurazione S3 Intelligent-Tiering su un bucket

L'`put-bucket-intelligent-tiering-configuration` esempio seguente aggiorna una configurazione S3 Intelligent-Tiering, denominata, su un bucket. `ExampleConfig` La configurazione trasferirà gli oggetti a cui non è stato effettuato l'accesso con le immagini del prefisso ad Archive Access dopo 90 giorni e Deep Archive Access dopo 180 giorni.

```
aws s3api put-bucket-intelligent-tiering-configuration \  
  --bucket DOC-EXAMPLE-BUCKET \  
  --id "ExampleConfig" \  
  --intelligent-tiering-configuration file://intelligent-tiering-  
configuration.json
```

Contenuto di `intelligent-tiering-configuration.json`.

```
{  
  "Id": "ExampleConfig",  
  "Status": "Enabled",  
  "Filter": {  
    "Prefix": "images"  
  },  
  "Tierings": [  
    {  
      "Days": 90,  
      "AccessTier": "ARCHIVE_ACCESS"  
    }  
  ]  
}
```

```
    },
    {
      "Days": 180,
      "AccessTier": "DEEP_ARCHIVE_ACCESS"
    }
  ]
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Setting Object Ownership su un bucket esistente](#) nella Amazon S3 User Guide.

- Per API i dettagli, consulta AWS CLI Command [PutBucketIntelligentTieringConfiguration](#) Reference.

## put-bucket-inventory-configuration

Il seguente esempio di codice mostra come utilizzare `put-bucket-inventory-configuration`.

### AWS CLI

Esempio 1: per impostare una configurazione di inventario per un bucket

L'`put-bucket-inventory-configuration` esempio seguente imposta un rapporto di inventario in ORC formato settimanale per il bucket. `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" } }'
```

Questo comando non produce alcun output.

Esempio 2: per impostare una configurazione di inventario per un bucket

L'`put-bucket-inventory-configuration` esempio seguente imposta un rapporto di inventario in CSV formato giornaliero per il bucket. `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" } }'
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutBucketInventoryConfiguration](#) in AWS CLI Command Reference.

## put-bucket-lifecycle-configuration

Il seguente esempio di codice mostra come utilizzare `put-bucket-lifecycle-configuration`.

### AWS CLI

Il comando seguente applica una configurazione del ciclo di vita a un bucket denominato: `my-bucket`

```
aws s3api put-bucket-lifecycle-configuration --bucket my-bucket --lifecycle-
configuration file:///lifecycle.json
```

Il file `lifecycle.json` è un JSON documento nella cartella corrente che specifica due regole:

```
{
  "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"
    }
]
}

```

La prima regola sposta i file con il prefisso `rotated` su Glacier nella data specificata. La seconda regola sposta le vecchie versioni degli oggetti su Glacier quando non sono più attuali. Per informazioni sui formati di timestamp accettabili, consultate [Specificazione dei valori dei parametri](#) nella Guida per l'utente.AWS CLI

- Per API i dettagli, vedere [PutBucketLifecycleConfiguration](#) in AWS CLI Command Reference.

## put-bucket-lifecycle

Il seguente esempio di codice mostra come utilizzare `put-bucket-lifecycle`.

### AWS CLI

Il comando seguente applica una configurazione del ciclo di vita al bucket: `my-bucket`

```
aws s3api put-bucket-lifecycle --bucket my-bucket --lifecycle-configuration file://lifecycle.json
```

Il file `lifecycle.json` è un JSON documento nella cartella corrente che specifica due regole:

```

{
  "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"
}
]
```

La prima regola sposta i file in Amazon Glacier dopo sessanta giorni. La seconda regola elimina i file da Amazon S3 alla data specificata. Per informazioni sui formati di timestamp accettabili, consulta [Specificazione dei valori dei parametri](#) nella Guida per l'utente AWS CLI.

Ogni regola dell'esempio precedente specifica una politica (`TransitionorExpiration`) e un prefisso di file (nome della cartella) a cui si applica. Puoi anche creare una regola che si applica a un intero bucket specificando un prefisso vuoto:

```
{
  "Rules": [
    {
      "ID": "Move to Glacier after sixty days (all objects in bucket)",
      "Prefix": "",
      "Status": "Enabled",
      "Transition": {
        "Days": 60,
        "StorageClass": "GLACIER"
      }
    }
  ]
}
```

- Per API i dettagli, vedere [PutBucketLifecycle](#) in AWS CLI Command Reference.

## put-bucket-logging

Il seguente esempio di codice mostra come utilizzare `put-bucket-logging`.

### AWS CLI

Esempio 1: per impostare la registrazione delle policy relative ai bucket

L'put-bucket-loggingesempio seguente imposta la politica di registrazione per. MyBucket Innanzitutto, concedi al servizio di registrazione l'autorizzazione principale nella tua policy del bucket utilizzando il comando. put-bucket-policy

```
aws s3api put-bucket-policy \  
  --bucket MyBucket \  
  --policy file://policy.json
```

Contenuto di 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"}  
      }  
    }  
  ]  
}
```

Per applicare la politica di registrazione, usa. put-bucket-logging

```
aws s3api put-bucket-logging \  
  --bucket MyBucket \  
  --bucket-logging-status file://logging.json
```

Contenuto di logging.json.

```
{  
  "LoggingEnabled": {  
    "TargetBucket": "MyBucket",  
    "TargetPrefix": "Logs/"  
  }  
}
```

Il `put-bucket-policy` comando è necessario per concedere `s3:PutObject` le autorizzazioni al responsabile del servizio di registrazione.

Per ulteriori informazioni, consulta [Amazon S3 Server Access Logging](#) nella Amazon S3 User Guide.

Esempio 2: impostare una policy sui bucket per registrare l'accesso a un solo utente

L'`put-bucket-logging` seguente imposta la politica di registrazione per `MyBucket`. L'AWS utente `bob@example.com` avrà il pieno controllo sui file di registro e nessun altro potrà accedervi. Innanzitutto, concedi l'autorizzazione a S3 con `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
```

Quindi applica la politica di registrazione utilizzando `put-bucket-logging`

```
aws s3api put-bucket-logging \  
  --bucket MyBucket \  
  --bucket-logging-status file://logging.json
```

Contenuto di `logging.json`.

```
{  
  "LoggingEnabled": {  
    "TargetBucket": "MyBucket",  
    "TargetPrefix": "MyBucketLogs/",  
    "TargetGrants": [  
      {  
        "Grantee": {  
          "Type": "AmazonCustomerByEmail",  
          "EmailAddress": "bob@example.com"  
        },  
        "Permission": "FULL_CONTROL"  
      }  
    ]  
  }  
}
```

il `put-bucket-acl` comando è necessario per concedere al sistema di consegna dei log di S3 le autorizzazioni necessarie (autorizzazioni di scrittura e lettura `acl`).

Per ulteriori informazioni, consulta [Amazon S3 Server Access Logging](#) nella Amazon S3 Developer Guide.

- Per API i dettagli, consulta Command [PutBucketLoggingReference](#) AWS CLI .

## **put-bucket-metrics-configuration**

Il seguente esempio di codice mostra come utilizzare `put-bucket-metrics-configuration`.

### AWS CLI

Per impostare una configurazione delle metriche per un bucket

L'`put-bucket-metrics-configuration` esempio seguente imposta una configurazione metrica con ID 123 per il bucket specificato.

```
aws s3api put-bucket-metrics-configuration \  
  --bucket my-bucket \  
  --id 123 \  
  --metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutBucketMetricsConfiguration](#) in AWS CLI Command Reference.

## **put-bucket-notification-configuration**

Il seguente esempio di codice mostra come utilizzare `put-bucket-notification-configuration`.

### AWS CLI

Per abilitare le notifiche specificate in un bucket

L'`put-bucket-notification-configuration` esempio seguente applica una configurazione di notifica a un bucket denominato `my-bucket`. Il file `notification.json` è un JSON documento nella cartella corrente che specifica un SNS argomento e un tipo di evento da monitorare.

```
aws s3api put-bucket-notification-configuration \  
  --bucket my-bucket \  
  --notification-configuration file://notification.json
```

Contenuto di notification.json.

```
{  
  "TopicConfigurations": [  
    {  
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic",  
      "Events": [  
        "s3:ObjectCreated:*"  
      ]  
    }  
  ]  
}
```

L'SNSargomento deve avere una IAM policy allegata che consenta ad Amazon S3 di pubblicare su di esso.

```
{  
  "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"  
        }  
      }  
    }  
  ]  
}
```

- Per API i dettagli, consulta [PutBucketNotificationConfiguration AWS CLI Command Reference](#).

## put-bucket-notification

Il seguente esempio di codice mostra come utilizzare `put-bucket-notification`.

### AWS CLI

Applica una configurazione di notifica a un bucket denominato `my-bucket`:

```
aws s3api put-bucket-notification --bucket my-bucket --notification-configuration file://notification.json
```

Il file `notification.json` è un JSON documento nella cartella corrente che specifica un SNS argomento e un tipo di evento da monitorare:

```
{
  "TopicConfiguration": {
    "Event": "s3:ObjectCreated:*",
    "Topic": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic"
  }
}
```

L'SNS argomento deve avere una IAM policy allegata che consenta ad Amazon S3 di pubblicare su di esso:

```
{
  "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"
  }
}
]
}
```

- Per API i dettagli, consulta [PutBucketNotification AWS CLI Command Reference](#).

## put-bucket-ownership-controls

Il seguente esempio di codice mostra come utilizzare `put-bucket-ownership-controls`.

### AWS CLI

Per aggiornare le impostazioni di proprietà del bucket di un bucket

L'`put-bucket-ownership-controls` esempio seguente aggiorna le impostazioni di proprietà del bucket di un bucket.

```
aws s3api put-bucket-ownership-controls \
  --bucket DOC-EXAMPLE-BUCKET \
  --ownership-controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Setting Object Ownership su un bucket esistente](#) nella Amazon S3 User Guide.

- Per API i dettagli, consulta AWS CLI Command [PutBucketOwnershipControls](#) Reference.

## put-bucket-policy

Il seguente esempio di codice mostra come utilizzare `put-bucket-policy`.

### AWS CLI

Questo esempio consente a tutti gli utenti di recuperare qualsiasi oggetto in MyBucket.

MySecretFolder Inoltre concede `put` e `delete` autorizzazione all'utente root dell' AWS account: 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/*"
    }
  ]
}

```

- Per API i dettagli, vedere [PutBucketPolicy](#) in AWS CLI Command Reference.

## put-bucket-replication

Il seguente esempio di codice mostra come utilizzare `put-bucket-replication`.

### AWS CLI

Per configurare la replica per un bucket S3

L'`put-bucket-replication` esempio seguente applica una configurazione di replica al bucket S3 specificato.



```
aws s3api put-bucket-replication \  
  --bucket AWSDOC-EXAMPLE-BUCKET1 \  
  --replication-configuration file://replication.json
```

Contenuto di 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"  
      }  
    }  
  ]  
}
```

Il bucket di destinazione deve avere il controllo delle versioni abilitato. Il ruolo specificato deve avere l'autorizzazione a scrivere nel bucket di destinazione e avere una relazione di trust che consenta ad Amazon S3 di assumere il ruolo.

Esempio di politica di autorizzazione dei ruoli:

```
{  
  "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/*"
  }
]
}

```

Esempio di politica sulle relazioni di fiducia:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Questo è il titolo dell'argomento](#) nella Guida per l'utente della console di Amazon Simple Storage Service.

- Per API i dettagli, consulta [PutBucketReplication AWS CLI Command Reference](#).

## put-bucket-request-payment

Il seguente esempio di codice mostra come utilizzare `put-bucket-request-payment`.

### AWS CLI

Esempio 1: abilitare la configurazione `requester pays` per un bucket

L'esempio seguente abilita il bucket specificato da `put-bucket-request-payment --requester pays`.

```
aws s3api put-bucket-request-payment \  
  --bucket my-bucket \  
  --request-payment-configuration '{"Payer":"Requester"}'
```

Questo comando non produce alcun output.

Esempio 2: disabilitare la configurazione `requester pays` per un bucket

L'esempio seguente disabilita per il bucket specificato da `put-bucket-request-payment --requester pays`.

```
aws s3api put-bucket-request-payment \  
  --bucket my-bucket \  
  --request-payment-configuration '{"Payer":"BucketOwner"}'
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutBucketRequestPayment](#) in AWS CLI Command Reference.

## put-bucket-tagging

Il seguente esempio di codice mostra come utilizzare `put-bucket-tagging`.

### AWS CLI

Il comando seguente applica una configurazione di tag a un bucket denominato: `my-bucket`

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging file://tagging.json
```

Il file `tagging.json` è un JSON documento nella cartella corrente che specifica i tag:

```
{
  "TagSet": [
    {
      "Key": "organization",
      "Value": "marketing"
    }
  ]
}
```

Oppure applica una configurazione di tagging `my-bucket` direttamente dalla riga di comando:

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging
'TagSet=[{Key=organization,Value=marketing}]'
```

- Per API i dettagli, consulta [PutBucketTagging AWS CLI Command Reference](#).

## put-bucket-versioning

Il seguente esempio di codice mostra come utilizzare `put-bucket-versioning`.

### AWS CLI

Il comando seguente abilita il controllo delle versioni su un bucket denominato: `my-bucket`

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-
configuration Status=Enabled
```

Il comando seguente abilita il controllo delle versioni e utilizza un codice mfa

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-
configuration Status=Enabled --mfa "SERIAL 123456"
```

- Per API i dettagli, vedere [PutBucketVersioning](#) in AWS CLI Command Reference.

## put-bucket-website

Il seguente esempio di codice mostra come utilizzare `put-bucket-website`.

### AWS CLI

Applica una configurazione statica del sito Web a un bucket denominato `my-bucket`:

```
aws s3api put-bucket-website --bucket my-bucket --website-configuration file://  
website.json
```

Il file `website.json` è un JSON documento nella cartella corrente che specifica l'indice e le pagine di errore per il sito Web:

```
{  
  "IndexDocument": {  
    "Suffix": "index.html"  
  },  
  "ErrorDocument": {  
    "Key": "error.html"  
  }  
}
```

- Per API i dettagli, vedere [PutBucketWebsite](#) in AWS CLI Command Reference.

## put-object-acl

Il seguente esempio di codice mostra come utilizzare `put-object-acl`.

### AWS CLI

Il comando seguente concede `full control` a due AWS utenti (`user1@example.com` e `user2@example.com`) `read` l'autorizzazione a tutti:

```
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
```

Vedi <http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html> per i dettagli sulla personalizzazione ACLs (i ACL comandi `s3api`, ad esempio `put-object-acl`, usano la stessa notazione abbreviata degli argomenti).

- Per i API dettagli, vedere in Command Reference. [PutObjectAcl](#) AWS CLI

## put-object-legal-hold

Il seguente esempio di codice mostra come utilizzare `put-object-legal-hold`.

## AWS CLI

Per applicare una conservazione a fini legali a un oggetto

L'`put-object-legal-hold` seguente imposta un Legal Hold sull'oggetto `doc1.rtf`.

```
aws s3api put-object-legal-hold \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf \  
  --legal-hold Status=ON
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutObjectLegalHold](#) in AWS CLI Command Reference.

## `put-object-lock-configuration`

Il seguente esempio di codice mostra come utilizzare `put-object-lock-configuration`.

## AWS CLI

Per impostare una configurazione di blocco degli oggetti su un bucket

L'`put-object-lock-configuration` seguente imposta un blocco degli oggetti di 50 giorni sul bucket specificato.

```
aws s3api put-object-lock-configuration \  
  --bucket my-bucket-with-object-lock \  
  --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule":  
  { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 } } }'
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutObjectLockConfiguration](#) in AWS CLI Command Reference.

## `put-object-retention`

Il seguente esempio di codice mostra come utilizzare `put-object-retention`.

## AWS CLI

Per impostare una configurazione di conservazione degli oggetti per un oggetto

L'`put-object-retention` esempio seguente imposta una configurazione di conservazione degli oggetti per l'oggetto specificato fino al 01/01/2025.

```
aws s3api put-object-retention \  
  --bucket my-bucket-with-object-lock \  
  --key doc1.rtf \  
  --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Questo comando non produce alcun output.

- Per i API dettagli, vedere in Command Reference. [PutObjectRetention](#) AWS CLI

## put-object-tagging

Il seguente esempio di codice mostra come utilizzare `put-object-tagging`.

### AWS CLI

Per impostare un tag su un oggetto

L'`put-object-tagging` esempio seguente imposta un tag con la chiave `designation` e il valore `confidential` sull'oggetto specificato.

```
aws s3api put-object-tagging \  
  --bucket my-bucket \  
  --key doc1.rtf \  
  --tagging '{"TagSet": [{ "Key": "designation", "Value": "confidential" } ]}'
```

Questo comando non produce alcun output.

L'`put-object-tagging` esempio seguente imposta più set di tag sull'oggetto specificato.

```
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" } ]}'
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutObjectTagging](#) in AWS CLI Command Reference.

## put-object

Il seguente esempio di codice mostra come utilizzare `put-object`.

### AWS CLI

L'esempio seguente utilizza il `put-object` comando per caricare un oggetto su Amazon S3:

```
aws s3api put-object --bucket text-content --key dir-1/my_images.tar.bz2 --  
body my_images.tar.bz2
```

L'esempio seguente mostra il caricamento di un file video (il file video viene specificato utilizzando la sintassi del file system Windows. ):

```
aws s3api put-object --bucket text-content --key dir-1/big-video-file.mp4 --body e:  
\media\videos\f-sharp-3-data-services.mp4
```

Per ulteriori informazioni sul caricamento di oggetti, consulta [Uploading Objects](#) nella Amazon S3 Developer Guide.

- Per API i dettagli, consulta [Command PutObject](#) Reference AWS CLI .

## put-public-access-block

Il seguente esempio di codice mostra come utilizzare `put-public-access-block`.

### AWS CLI

Per impostare la configurazione di accesso pubblico a blocchi per un bucket

L'`put-public-access-block` esempio seguente imposta una configurazione restrittiva di accesso pubblico a blocchi per il bucket specificato.

```
aws s3api put-public-access-block \  
  --bucket my-bucket \  
  --public-access-block-  
configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutPublicAccessBlock](#) in AWS CLI Command Reference.



## rb

Il seguente esempio di codice mostra come utilizzare `rb`.

### AWS CLI

#### Esempio 1: eliminare un bucket

Il `rb` comando seguente rimuove un bucket. In questo esempio, il bucket dell'utente è `mybucket`. Nota che il secchio deve essere vuoto per rimuovere:

```
aws s3 rb s3://mybucket
```

Output:

```
remove_bucket: mybucket
```

#### Esempio 2: eliminazione forzata di un bucket

Il `rb` comando seguente utilizza il `--force` parametro per rimuovere prima tutti gli oggetti nel bucket e poi rimuovere il bucket stesso. In questo esempio, il bucket dell'utente è `mybucket` e gli oggetti in `mybucket` esso contenuti sono `e: test1.txt test2.txt`

```
aws s3 rb s3://mybucket \  
--force
```

Output:

```
delete: s3://mybucket/test1.txt  
delete: s3://mybucket/test2.txt  
remove_bucket: mybucket
```

- Per API i dettagli, consulta [Rb](#) in AWS CLI Command Reference.

## restore-object

Il seguente esempio di codice mostra come usare `restore-object`.

### AWS CLI

Per creare una richiesta di ripristino per un oggetto

L'`restore-object` seguente ripristina l'oggetto Amazon S3 Glacier specificato per il `my-glacier-bucket` bucket per 10 giorni.

```
aws s3api restore-object \  
  --bucket my-glacier-bucket \  
  --key doc1.rtf \  
  --restore-request Days=10
```

Questo comando non produce alcun output.

- Per i API dettagli, consulta Command Reference. [RestoreObject](#) AWS CLI

## **rm**

Il seguente esempio di codice mostra come utilizzare `rm`.

### AWS CLI

Esempio 1: eliminare un oggetto S3

Il `rm` comando seguente elimina un singolo oggetto s3:

```
aws s3 rm s3://mybucket/test2.txt
```

Output:

```
delete: s3://mybucket/test2.txt
```

Esempio 2: elimina tutti i contenuti in un bucket

Il `rm` comando seguente elimina in modo ricorsivo tutti gli oggetti contenuti in un bucket e in un prefisso specificati quando viene passato con il parametro. `--recursive` In questo esempio, il `mybucket` bucket contiene gli oggetti e: `test1.txt` `test2.txt`

```
aws s3 rm s3://mybucket \  
  --recursive
```

Output:

```
delete: s3://mybucket/test1.txt  
delete: s3://mybucket/test2.txt
```

### Esempio 3: Eliminare tutti i contenuti in un bucket, ad eccezione dei file ``.jpg``

Il `rm` comando seguente elimina in modo ricorsivo tutti gli oggetti contenuti in un bucket e prefisso specificati quando viene passato con il parametro, escludendo alcuni oggetti utilizzando un parametro. `--recursive --exclude` In questo esempio, il `mybucket` bucket contiene gli oggetti e: `test1.txt test2.jpg`

```
aws s3 rm s3://mybucket/ \  
  --recursive \  
  --exclude "*.jpg"
```

Output:

```
delete: s3://mybucket/test1.txt
```

### Esempio 4: elimina tutti i contenuti in un bucket, ad eccezione degli oggetti con il prefisso specificato

Il `rm` comando seguente elimina in modo ricorsivo tutti gli oggetti in un bucket e prefisso specificati quando viene passato con il parametro, `--recursive` mentre esclude tutti gli oggetti con un particolare prefisso utilizzando un parametro. `--exclude` In questo esempio, il bucket contiene gli oggetti e: `mybucket test1.txt another/test.txt`

```
aws s3 rm s3://mybucket/ \  
  --recursive \  
  --exclude "another/*"
```

Output:

```
delete: s3://mybucket/test1.txt
```

### Esempio 5: eliminare un oggetto da un punto di accesso S3

Il `rm` comando seguente elimina un singolo oggetto (`mykey`) dal punto di accesso (`myaccesspoint`). :: Il `rm` comando seguente elimina un singolo oggetto (`mykey`) dal punto di accesso (`myaccesspoint`).

```
aws s3 rm s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

Output:

```
delete: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

- Per API i dettagli, vedere [Rm](#) in AWS CLI Command Reference.

## select-object-content

Il seguente esempio di codice mostra come usare `select-object-content`.

### AWS CLI

Per filtrare il contenuto di un oggetto Amazon S3 in base a un'istruzione SQL

L'esempio seguente filtra l'oggetto `my-data-file.csv` con l'istruzione SQL specificata e invia l'output a un file.

```
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
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [SelectObjectContent](#) in AWS CLI Command Reference.

## sync

Il seguente esempio di codice mostra come utilizzare `sync`.

### AWS CLI

Esempio 1: sincronizza tutti gli oggetti locali con il bucket specificato

Il `sync` comando seguente sincronizza gli oggetti da una directory locale al prefisso e al bucket specificati caricando i file locali su S3. Un file locale richiederà il caricamento se la dimensione del file locale è diversa dalla dimensione dell'oggetto S3, l'ora dell'ultima modifica del file locale è successiva all'ora dell'ultima modifica dell'oggetto S3 o il file locale non esiste nel bucket e nel prefisso specificati. In questo esempio, l'utente sincronizza il bucket con la directory corrente

locale. mybucket La directory corrente locale contiene i file `test.txt` e `test2.txt` Il bucket non mybucket contiene oggetti.

```
aws s3 sync . s3://mybucket
```

Output:

```
upload: test.txt to s3://mybucket/test.txt
upload: test2.txt to s3://mybucket/test2.txt
```

Esempio 2: sincronizza tutti gli oggetti S3 dal bucket S3 specificato con un altro bucket

Il `sync` comando seguente sincronizza gli oggetti con un prefisso e un bucket specificati con gli oggetti con un altro prefisso e bucket specificati copiando gli oggetti S3. Un oggetto S3 richiederà la copia se le dimensioni dei due oggetti S3 sono diverse, l'ora dell'ultima modifica dell'origine è successiva all'ora dell'ultima modifica della destinazione o l'oggetto S3 non esiste nella destinazione del bucket e del prefisso specificati.

In questo esempio, l'utente sincronizza il bucket con il bucket. mybucket mybucket2 Il bucket mybucket contiene gli oggetti e. `test.txt` `test2.txt` Il bucket non mybucket2 contiene oggetti:

```
aws s3 sync s3://mybucket s3://mybucket2
```

Output:

```
copy: s3://mybucket/test.txt to s3://mybucket2/test.txt
copy: s3://mybucket/test2.txt to s3://mybucket2/test2.txt
```

Esempio 3: sincronizza tutti gli oggetti S3 dal bucket S3 specificato alla directory locale

Il `sync` comando seguente sincronizza i file dal bucket S3 specificato alla directory locale scaricando oggetti S3. Un oggetto S3 richiederà il download se la dimensione dell'oggetto S3 è diversa dalla dimensione del file locale, l'ora dell'ultima modifica dell'oggetto S3 è più recente dell'ora dell'ultima modifica del file locale o l'oggetto S3 non esiste nella directory locale. Tieni presente che quando gli oggetti vengono scaricati da S3, l'ora dell'ultima modifica del file locale viene sostituita dall'ora dell'ultima modifica dell'oggetto S3. In questo esempio, l'utente sincronizza il bucket con la directory locale mybucket corrente. Il bucket mybucket contiene gli oggetti e. `test.txt` `test2.txt` La directory locale corrente non contiene file:

```
aws s3 sync s3://mybucket .
```

Output:

```
download: s3://mybucket/test.txt to test.txt
download: s3://mybucket/test2.txt to test2.txt
```

Esempio 4: sincronizza tutti gli oggetti locali nel bucket specificato ed elimina tutti i file che non corrispondono

Il sync comando seguente sincronizza gli oggetti con un prefisso e un bucket specificati con i file in una directory locale caricando i file locali su S3. A causa del `--delete` parametro, tutti i file esistenti con il prefisso e il bucket specificati ma non esistenti nella directory locale verranno eliminati. In questo esempio, l'utente sincronizza il bucket con la directory corrente `mybucket` locale. La directory corrente locale contiene i file `test.txt` e `test2.txt`. Il bucket `mybucket` contiene l'oggetto `test3.txt`:

```
aws s3 sync . s3://mybucket \
--delete
```

Output:

```
upload: test.txt to s3://mybucket/test.txt
upload: test2.txt to s3://mybucket/test2.txt
delete: s3://mybucket/test3.txt
```

Esempio 5: sincronizza tutti gli oggetti locali con il bucket specificato tranne i file `*.jpg`

Il sync comando seguente sincronizza gli oggetti con un prefisso e un bucket specificati con i file in una directory locale caricando i file locali su S3. A causa del `--exclude` parametro, tutti i file che corrispondono allo schema esistente sia in S3 che localmente verranno esclusi dalla sincronizzazione. In questo esempio, l'utente sincronizza il bucket con la `mybucket` directory corrente locale. La directory corrente locale contiene i file `test.jpg` e `test2.txt`. Il bucket `mybucket` contiene l'oggetto `test.jpg` di dimensioni diverse da quelle locali `test.jpg`:

```
aws s3 sync . s3://mybucket \
--exclude "*.jpg"
```

Output:

```
upload: test2.txt to s3://mybucket/test2.txt
```

Esempio 6: sincronizza tutti gli oggetti locali con il bucket specificato tranne i file ``.jpg``

Il sync comando seguente sincronizza i file in una directory locale con gli oggetti con un prefisso e un bucket specificati scaricando oggetti S3. Questo esempio utilizza il `--exclude` parametro flag per escludere una directory e un prefisso S3 specificati dal comando. sync In questo esempio, l'utente sincronizza la directory corrente locale con il bucket. mybucket La directory corrente locale contiene i file `test.txt` e `another/test2.txt` Il bucket mybucket contiene gli oggetti `another/test5.txt` e `etest1.txt`:

```
aws s3 sync s3://mybucket/ . \
--exclude "*another/*"
```

Output:

```
download: s3://mybucket/test1.txt to test1.txt
```

Esempio 7: sincronizza tutti gli oggetti tra i bucket in diverse regioni

Il sync comando seguente sincronizza i file tra due bucket in regioni diverse:

```
aws s3 sync s3://my-us-west-2-bucket s3://my-us-east-1-bucket \
--source-region us-west-2 \
--region us-east-1
```

Output:

```
download: s3://my-us-west-2-bucket/test1.txt to s3://my-us-east-1-bucket/test1.txt
```

Esempio 8: sincronizzazione con un punto di accesso S3

Il sync comando seguente sincronizza la directory corrente con il punto di accesso (myaccesspoint):

```
aws s3 sync . s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

Output:

```
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
```

- Per API i dettagli, vedete [Sync](#) in AWS CLI Command Reference.

## upload-part-copy

Il seguente esempio di codice mostra come utilizzare `upload-part-copy`.

### AWS CLI

Caricare parte di un oggetto copiando i dati da un oggetto esistente come origine dati

L'esempio seguente carica una parte copiando i dati da un oggetto esistente come origine dati.

```
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_RiBy0gg0H6b13pVRY7QjvL1f75iFdJqp_2wztk5hvpUM2SesXgrzbeh"
```

Output:

```
{
  "CopyPartResult": {
    "LastModified": "2019-12-13T23:16:03.000Z",
    "ETag": "\"711470fc377698c393d94aed6305e245\""
  }
}
```

- Per API i dettagli, vedere [UploadPartCopy](#) in AWS CLI Command Reference.

## upload-part

Il seguente esempio di codice mostra come utilizzare `upload-part`.



## AWS CLI

Il comando seguente carica la prima parte di un caricamento in più parti avviato con il comando: `create-multipart-upload`

```
aws s3api upload-part --bucket my-bucket --key 'multipart/01' --part-number 1 --  
body part01 --upload-id  
"dfRtDYU0WCCcH43C3WFbkR0NycyCpTJJvxu2i5GYkZlJF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR"
```

L'opzione `body` richiede il nome o il percorso di un file locale per il caricamento (non utilizzate il prefisso `file://`). La dimensione minima della parte è di 5 MB. L'ID di caricamento viene restituito da `create-multipart-upload` e può essere recuperato anche con `list-multipart-uploads`. Bucket e chiave vengono specificati quando si crea il caricamento multiparte.

Output:

```
{  
  "ETag": "\"e868e0f4719e394144ef36531ee6824c\""  
}
```

Salva il ETag valore di ogni parte per utilizzarlo in un secondo momento. Sono necessari per completare il caricamento in più parti.

- Per API i dettagli, vedere [UploadPart](#) in AWS CLI Command Reference.

## website

Il seguente esempio di codice mostra come utilizzare `website`.

## AWS CLI

Configura un bucket S3 come sito Web statico

Il comando seguente configura un bucket `my-bucket` denominato sito Web statico. L'opzione `index document` specifica il file a `my-bucket` cui verranno indirizzati i visitatori quando accedono al sito Web. URL In questo caso, il bucket si trova nella regione `us-west-2`, quindi il sito apparirà in `http://my-bucket.s3-website-us-west-2.amazonaws.com`

Tutti i file nel bucket che appaiono nel sito statico devono essere configurati per consentire ai visitatori di aprirli. Le autorizzazioni relative ai file sono configurate separatamente dalla configurazione del sito Web del bucket.

```
aws s3 website s3://my-bucket/ \  
  --index-document index.html \  
  --error-document error.html
```

Per informazioni sull'hosting di un sito Web statico in Amazon S3, consulta [Hosting a Static Website](#) nella Amazon Simple Storage Service Developer Guide.

- Per API i dettagli, consulta [Website](#) in AWS CLI Command Reference.

## Esempi di Amazon S3 Control con AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il controllo AWS Command Line Interface con Amazon S3.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **create-access-point**

Il seguente esempio di codice mostra come utilizzare `create-access-point`.

#### AWS CLI

Per creare un punto di accesso

L'esempio seguente crea un punto di accesso denominato `finance-ap` in base al bucket `business-records` nell'account `123456789012`. Prima di eseguire questo esempio, sostituisci il nome del punto di accesso, il nome del bucket e il numero di account con i valori appropriati per il tuo caso d'uso.

```
aws s3control create-access-point \  
  --bucket business-records \  
  --access-point-name finance-ap \  
  --account-id 123456789012
```

```
--account-id 123456789012 \  
--bucket business-records \  
--name finance-ap
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Creating Access Points](#) nella Amazon Simple Storage Service Developer Guide.

- Per API i dettagli, consulta [CreateAccessPoint AWS CLI Command Reference](#).

## create-job

Il seguente esempio di codice mostra come utilizzare `create-job`.

### AWS CLI

Per creare un processo di operazioni in batch su Amazon S3

L'esempio seguente crea un processo di operazioni batch di Amazon S3 con cui etichettare gli oggetti `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
```

Output:

```
{  
  "JobId": "93735294-df46-44d5-8638-6356f335324e"  
}
```

- Per API i dettagli, consulta [CreateJob AWS CLI Command Reference](#).

## delete-access-point-policy

Il seguente esempio di codice mostra come utilizzare `delete-access-point-policy`.

### AWS CLI

Per eliminare una policy relativa ai punti di accesso

L'esempio seguente elimina la policy del punto di accesso dal punto di accesso denominato `finance-ap` nell'account `123456789012`. Prima di eseguire questo esempio, sostituite il nome del punto di accesso e il numero di account con i valori appropriati per il vostro caso d'uso.

```
aws s3control delete-access-point-policy \  
  --account-id 123456789012 \  
  --name finance-ap
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing Data Access with Amazon S3 Access Points](#) nella Amazon Simple Storage Service Developer Guide.

- Per API i dettagli, consulta [DeleteAccessPointPolicy AWS CLI Command Reference](#).

## delete-access-point

Il seguente esempio di codice mostra come utilizzare `delete-access-point`.

### AWS CLI

Per eliminare un punto di accesso

L'esempio seguente elimina un punto di accesso denominato `finance-ap` nell'account `123456789012`. Prima di eseguire questo esempio, sostituite il nome del punto di accesso e il numero di account con i valori appropriati per il vostro caso d'uso.

```
aws s3control delete-access-point \  
  --account-id 123456789012 \  
  --name finance-ap
```

```
--name finance-ap
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing Data Access with Amazon S3 Access Points](#) nella Amazon Simple Storage Service Developer Guide.

- Per API i dettagli, consulta [DeleteAccessPoint AWS CLI Command Reference](#).

## **delete-public-access-block**

Il seguente esempio di codice mostra come utilizzare `delete-public-access-block`.

### AWS CLI

Per eliminare le impostazioni di blocco dell'accesso pubblico per un account

L'`delete-public-access-block` esempio seguente elimina le impostazioni di blocco dell'accesso pubblico per l'account specificato.

```
aws s3control delete-public-access-block \  
  --account-id 123456789012
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeletePublicAccessBlock](#) in AWS CLI Command Reference.

## **describe-job**

Il seguente esempio di codice mostra come utilizzare `describe-job`.

### AWS CLI

Per descrivere un processo operativo in batch di Amazon S3

Di seguito vengono `describe-job` forniti i parametri di configurazione e lo stato per il processo di operazioni batch specificato.

```
aws s3control describe-job \  
  --account-id 123456789012 \  
  --job-id 93735294-df46-44d5-8638-6356f335324e
```

## Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [DescribeJob](#) in AWS CLI Command Reference.

## get-access-point-policy-status

Il seguente esempio di codice mostra come utilizzare `get-access-point-policy-status`.

### AWS CLI

Per recuperare lo stato della politica del punto di accesso

L'`get-access-point-policy-status` esempio seguente recupera lo stato della policy del punto di accesso per il punto di accesso denominato `finance-ap` nell'account `123456789012`. Lo stato della politica del punto di accesso indica se la politica del punto di accesso consente l'accesso pubblico. Prima di eseguire questo esempio, sostituite il nome del punto di accesso e il numero di account con valori appropriati per il vostro caso d'uso.

```
aws s3control get-access-point-policy-status \
  --account-id 123456789012 \
  --name finance-ap
```

Output:

```
{
  "PolicyStatus": {
    "IsPublic": false
  }
}
```

Per ulteriori informazioni su quando una policy di access point è considerata pubblica, consulta [|| significato di «pubblico»](#) nella Amazon Simple Storage Service Developer Guide.

- Per API i dettagli, consulta [GetAccessPointPolicyStatus AWS CLI](#) Command Reference.

## get-access-point-policy

Il seguente esempio di codice mostra come utilizzare `get-access-point-policy`.

### AWS CLI

Per recuperare una politica del punto di accesso

L'esempio seguente recupera la policy del punto di accesso dal punto di accesso denominato `finance-ap` nell'account `123456789012`. Prima di eseguire questo esempio, sostituite il nome del punto di accesso e il numero di account con i valori appropriati per il vostro caso d'uso.

```
aws s3control get-access-point-policy \
  --account-id 123456789012 \
  --name finance-ap
```

Output:

```
{
  "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/*\"}]}"
}
```

Per ulteriori informazioni, consulta [Managing Data Access with Amazon S3 Access Points](#) nella Amazon Simple Storage Service Developer Guide.

- Per API i dettagli, consulta [GetAccessPointPolicy AWS CLI Command Reference](#).

## get-access-point

Il seguente esempio di codice mostra come utilizzare `get-access-point`.

### AWS CLI

Per recuperare i dettagli di configurazione del punto di accesso

L'esempio seguente recupera i dettagli di configurazione per il punto di accesso denominato `finance-ap` nell'account `123456789012`. Prima di eseguire questo



esempio, sostituite il nome del punto di accesso e il numero di account con i valori appropriati per il vostro caso d'uso.

```
aws s3control get-access-point \  
  --account-id 123456789012 \  
  --name finance-ap
```

Output:

```
{  
  "Name": "finance-ap",  
  "Bucket": "business-records",  
  "NetworkOrigin": "Internet",  
  "PublicAccessBlockConfiguration": {  
    "BlockPublicAcls": false,  
    "IgnorePublicAcls": false,  
    "BlockPublicPolicy": false,  
    "RestrictPublicBuckets": false  
  },  
  "CreationDate": "2020-01-01T00:00:00Z"  
}
```

Per ulteriori informazioni, consulta [Managing Data Access with Amazon S3 Access Points](#) nella Amazon Simple Storage Service Developer Guide.

- Per API i dettagli, consulta [GetAccessPoint AWS CLI Command Reference](#).

## get-multi-region-access-point-routes

Il seguente esempio di codice mostra come utilizzare `get-multi-region-access-point-routes`.

AWS CLI

Per interrogare la configurazione corrente del percorso del punto di accesso multiregionale

L'`get-multi-region-access-point-routes` esempio seguente restituisce la configurazione di routing corrente per il punto di accesso multiregionale specificato.

```
aws s3control get-multi-region-access-point-routes \  
  --region Region \  
  --account-id 111122223333 \  
  --access-point ap-123456789012
```

```
--mrap MultiRegionAccessPoint_ARN
```

Output:

```
{
  "Mrap": "arn:aws:s3::111122223333:accesspoint/00000000000000000000.mrap",
  "Routes": [
    {
      "Bucket": "DOC-EXAMPLE-BUCKET-1",
      "Region": "ap-southeast-2",
      "TrafficDialPercentage": 100
    },
    {
      "Bucket": "DOC-EXAMPLE-BUCKET-2",
      "Region": "us-west-1",
      "TrafficDialPercentage": 0
    }
  ]
}
```

- Per API i dettagli, vedere [GetMultiRegionAccessPointRoutes](#) in AWS CLI Command Reference.

## get-public-access-block

Il seguente esempio di codice mostra come utilizzare `get-public-access-block`.

AWS CLI

Per elencare le impostazioni di accesso pubblico ai blocchi per un account

L'`get-public-access-block` esempio seguente visualizza le impostazioni di blocco dell'accesso pubblico per l'account specificato.

```
aws s3control get-public-access-block \
  --account-id 123456789012
```

Output:

```
{
  "PublicAccessBlockConfiguration": {
    "BlockPublicPolicy": true,
    "RestrictPublicBuckets": true,
  }
}
```

```
    "IgnorePublicAcls": true,  
    "BlockPublicAcls": true  
  }  
}
```

- Per API i dettagli, vedere [GetPublicAccessBlock](#) in AWS CLI Command Reference.

## list-access-points

Il seguente esempio di codice mostra come utilizzare `list-access-points`.

### AWS CLI

Esempio 1: per recuperare un elenco di tutti i punti di accesso per un account

L'`list-access-points` esempio seguente visualizza un elenco di tutti i punti di accesso collegati ai bucket di proprietà dell'account 123456789012.

```
aws s3control list-access-points \  
  --account-id 123456789012
```

Output:

```
{  
  "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"
    }
  ]
}
```

Esempio 2: per recuperare un elenco di tutti i punti di accesso per un bucket

L'`list-access-points` seguente recupera un elenco di tutti i punti di accesso collegati al bucket di `external-docs` proprietà dell'account `123456789012`.

```
aws s3control list-access-points \
  --account-id 123456789012 \
  --bucket external-docs
```

Output:

```
{
  "AccessPointList": [
    {
      "Name": "customer-ap",
      "NetworkOrigin": "Internet",
      "Bucket": "external-docs"
    },
    {
      "Name": "public-ap",
      "NetworkOrigin": "Internet",
      "Bucket": "external-docs"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Managing Data Access with Amazon S3 Access Points](#) nella Amazon Simple Storage Service Developer Guide.

- Per API i dettagli, consulta [ListAccessPoints AWS CLI Command Reference](#).

## list-jobs

Il seguente esempio di codice mostra come utilizzare `list-jobs`.

### AWS CLI

Per elencare un account, operazioni in batch (Amazon S3), processi operativi

L'`list-jobs` seguente elenca tutti i processi di operazioni batch recenti per l'account specificato.

```
aws s3control list-jobs \  
  --account-id 123456789012
```

Output:

```
{  
  "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
      },
    ]
  }

```

- Per API i dettagli, vedere [ListJobs](#) in AWS CLI Command Reference.

## put-access-point-policy

Il seguente esempio di codice mostra come utilizzare `put-access-point-policy`.

### AWS CLI

Per impostare una politica del punto di accesso

L'esempio seguente inserisce la politica del punto di accesso specificata per il punto di accesso `finance-ap` nell'account `123456789012`. Se il punto di accesso dispone già di una politica, questo comando sostituisce la politica esistente con quella specificata in questo comando. Prima di eseguire questo esempio, sostituite il numero di account, il nome del punto di accesso e le istruzioni relative alle policy con valori appropriati per il vostro caso d'uso.

```

aws s3control put-access-point-policy \
  --account-id 123456789012 \
  --name finance-ap \
  --policy file://ap-policy.json

```

Contenuto di `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/*"
    }
  ]
}

```

```
}
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing Data Access with Amazon S3 Access Points](#) nella Amazon Simple Storage Service Developer Guide.

- Per API i dettagli, consulta [PutAccessPointPolicy AWS CLI Command Reference](#).

## **put-public-access-block**

Il seguente esempio di codice mostra come utilizzare `put-public-access-block`.

### AWS CLI

Per modificare le impostazioni di blocco dell'accesso pubblico per un account

L'`put-public-access-block` esempio seguente attiva tutte le impostazioni di blocco dell'accesso pubblico `true` per l'account specificato.

```
aws s3control put-public-access-block \
  --account-id 123456789012 \
  --public-access-block-configuration '{"BlockPublicAcls": true,
  "IgnorePublicAcls": true, "BlockPublicPolicy": true, "RestrictPublicBuckets":
  true}'
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutPublicAccessBlock](#) in AWS CLI Command Reference.

## **submit-multi-region-access-point-routes**

Il seguente esempio di codice mostra come utilizzare `submit-multi-region-access-point-routes`.

### AWS CLI

Per aggiornare la configurazione del routing del punto di accesso multiregionale

L'`submit-multi-region-access-point-routes` esempio seguente aggiorna gli stati di routing di `DOC-EXAMPLE-BUCKET-1` e `DOC-EXAMPLE-BUCKET-2` nella `ap-southeast-2` regione per il punto di accesso multiregionale.

```
aws s3control submit-multi-region-access-point-routes \  
  --region ap-southeast-2 \  
  --account-id 111122223333 \  
  --mrarp MultiRegionAccessPoint_ARN \  
  --route-updates Bucket=DOC-EXAMPLE-  
BUCKET-1,TrafficDialPercentage=100 Bucket=DOC-EXAMPLE-  
BUCKET-2,TrafficDialPercentage=0
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [SubmitMultiRegionAccessPointRoutes](#) in AWS CLI Command Reference.

## update-job-priority

Il seguente esempio di codice mostra come utilizzare `update-job-priority`.

AWS CLI

Per aggiornare la priorità del processo di un processo di operazioni in batch di Amazon S3

L'`update-job-priority` esempio seguente aggiorna il lavoro specificato con una nuova priorità.

```
aws s3control update-job-priority \  
  --account-id 123456789012 \  
  --job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \  
  --priority 52
```

Output:

```
{  
  "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386",  
  "Priority": 52  
}
```

- Per API i dettagli, vedere [UpdateJobPriority](#) in AWS CLI Command Reference.

## update-job-status

Il seguente esempio di codice mostra come utilizzare `update-job-status`.



## AWS CLI

Per aggiornare lo stato di un processo operativo in batch di Amazon S3

L'update-job-statusesempio seguente annulla il lavoro specificato che è in attesa di approvazione.

```
aws s3control update-job-status \  
  --account-id 123456789012 \  
  --job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \  
  --requested-job-status Cancelled
```

Output:

```
{  
  "Status": "Cancelled",  
  "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386"  
}
```

L'update-job-statusesempio seguente conferma ed esegue il file specificato che è in attesa di approvazione.

```
aws s3control update-job-status \  
  --account-id 123456789012 \  
  --job-id 5782949f-3301-4fb3-be34-8d5bab54dbca \  
  --requested-job-status Ready
```

**Output::**

```
{  
  "Status": "Ready",  
  "JobId": "5782949f-3301-4fb3-  
be34-8d5bab54dbca"  
}
```

L'update-job-statusesempio seguente annulla il processo specificato che è in esecuzione.

```
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"
}
```

- Per API i dettagli, vedere [UpdateJobStatus](#) in AWS CLI Command Reference.

## Esempi di utilizzo di S3 Glacier AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando S3 Glacier AWS Command Line Interface .

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **abort-multipart-upload**

Il seguente esempio di codice mostra come utilizzare `abort-multipart-upload`.

#### AWS CLI

Il comando seguente elimina un caricamento multiparte in corso su un archivio denominato: `my-vault`

```
aws glacier abort-multipart-upload --account-id - --vault-name my-vault
--upload-id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-0ssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ
```

Il comando non produce output. Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso. L'ID di

caricamento viene restituito dal comando `aws glacier initiate-multipart-upload` e può essere ottenuto anche utilizzando `aws glacier list-multipart-uploads`.

Per ulteriori informazioni sui caricamenti multiparte su Amazon Glacier utilizzando Amazon Glacier, AWS CLI consulta [Using Amazon Glacier](#) nella Guida per l'utente AWS CLI

- Per i API dettagli, consulta Command Reference. [AbortMultipartUpload](#) AWS CLI

## abort-vault-lock

Il seguente esempio di codice mostra come utilizzare `abort-vault-lock`.

AWS CLI

Per interrompere un processo di blocco del vault in corso

L'esempio seguente elimina una politica di blocco del vault dal vault specificato e reimposta lo stato di blocco del vault su sbloccato.

```
aws glacier abort-vault-lock \  
  --account-id - \  
  --vault-name MyVaultName
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Abort Vault Lock \(DELETElock-policy\)](#) nella Amazon Glacier Developer Guide. API

- Per i API dettagli, consulta Command Reference. [AbortVaultLock](#) AWS CLI

## add-tags-to-vault

Il seguente esempio di codice mostra come utilizzare `add-tags-to-vault`.

AWS CLI

Il comando seguente aggiunge due tag a un vault denominato `my-vault`:

```
aws glacier add-tags-to-vault --account-id - --vault-name my-vault --  
tags id=1234,date=july2015
```

Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

- Per API i dettagli, vedere [AddTagsToVault](#) in AWS CLI Command Reference.

## complete-multipart-upload

Il seguente esempio di codice mostra come utilizzare `complete-multipart-upload`.

### AWS CLI

Il comando seguente completa il caricamento in più parti per un archivio da 3 MiB:

```
aws glacier complete-multipart-upload --archive-size 3145728 --  
checksum 9628195fcdcbbbe76cdde456d4646fa7de5f219fb39823836d81f0cc0e18aa67  
--upload-id 19gaRezEXAMPLES6Ry5YYdqthH0C_kGRCT03L9yetr220UmPtBYKk-  
0ssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ --account-id - --vault-name my-vault
```

Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

L'ID di caricamento viene restituito dal comando `aws glacier initiate-multipart-upload` e può essere ottenuto anche utilizzando `aws glacier list-multipart-uploads`. Il parametro `checksum` richiede un hash ad albero SHA -256 dell'archivio in formato esadecimale.

Per ulteriori informazioni sui caricamenti multiparte su Amazon Glacier utilizzando, comprese le istruzioni sul calcolo di un AWS CLI tree hash, consulta [Using Amazon Glacier](#) nella Guida per l'utente. AWS CLI

- Per i dettagli, consulta Command Reference. API [CompleteMultipartUpload](#) AWS CLI

## complete-vault-lock

Il seguente esempio di codice mostra come utilizzare `complete-vault-lock`.

### AWS CLI

Per completare un processo di blocco del vault in corso

L'esempio seguente completa l'avanzamento del blocco in corso per l'archivio specificato e imposta lo stato di blocco del deposito su `Locked`. Si ottiene il valore del parametro durante l'esecuzione di `initiate-lock-process`

```
aws glacier complete-vault-lock \  
  --account-id - \  
  --vault-name MyVaultName \  
  --lock-id 9QZgEXAMPLEPhvL6xEXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Complete Vault Lock \(POSTlockId\)](#) nella Amazon Glacier Developer API Guide.

- Per API i dettagli, consulta Command [CompleteVaultLock](#)Reference AWS CLI .

## create-vault

Il seguente esempio di codice mostra come utilizzare `create-vault`.

### AWS CLI

Il comando seguente crea un nuovo vault denominato `my-vault`:

```
aws glacier create-vault --vault-name my-vault --account-id -
```

Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

- Per API i dettagli, vedere [CreateVault](#)in AWS CLI Command Reference.

## delete-archive

Il seguente esempio di codice mostra come utilizzare `delete-archive`.

### AWS CLI

Eliminazione di un archivio da un vault

L'esempio `delete-archive` seguente rimuove l'archivio specificato da `example_vault`.

```
aws glacier delete-archive \  
  --account-id 111122223333 \  
  --vault-name example_vault \  
  --archive-id EXAMPLE
```

```
--archive-id Sc0u9ZP8yaWkmh-XGLIvAVprtLhaLCGnNwNL5I5x9HqPIkX5mjc0DrId3Ln-Gi_k2HzmLIDZUz117KSdVMdMXLuFwi9PJUitxW073edQ43eTLMWkH0pd9zVSAuV_XXZBVhKhyGhJ7w
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteArchive](#) in AWS CLI Command Reference.

## delete-vault-access-policy

Il seguente esempio di codice mostra come utilizzare `delete-vault-access-policy`.

### AWS CLI

Per rimuovere la politica di accesso di un vault

L'`delete-vault-access-policy` esempio seguente rimuove la politica di accesso per il vault specificato.

```
aws glacier delete-vault-access-policy \  
  --account-id 111122223333 \  
  --vault-name example_vault
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteVaultAccessPolicy](#) in AWS CLI Command Reference.

## delete-vault-notifications

Il seguente esempio di codice mostra come utilizzare `delete-vault-notifications`.

### AWS CLI

Per rimuovere le SNS notifiche per un vault

L'`delete-vault-notifications` esempio seguente rimuove le notifiche inviate da Amazon Simple Notification Service (Amazon SNS) per il vault specificato.

```
aws glacier delete-vault-notifications \  
  --account-id 111122223333 \  
  --vault-name example_vault
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [DeleteVaultNotifications AWS CLI Command Reference](#).

## delete-vault

Il seguente esempio di codice mostra come utilizzare `delete-vault`.

### AWS CLI

Il comando seguente elimina un vault denominato `my-vault`:

```
aws glacier delete-vault --vault-name my-vault --account-id -
```

Il comando non produce output. Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

- Per API i dettagli, vedere [DeleteVault](#) in AWS CLI Command Reference.

## describe-job

Il seguente esempio di codice mostra come utilizzare `describe-job`.

### AWS CLI

Il comando seguente recupera le informazioni su un processo di recupero dell'inventario su un vault denominato `my-vault`:

```
aws glacier describe-job --account-id - --vault-name my-vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-R047Yc6FxsdsGBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW
```

Output:

```
{
  "InventoryRetrievalParameters": {
    "Format": "JSON"
  },
  "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
  "Completed": false,
  "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-R047Yc6FxsdsGBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW",
  "Action": "InventoryRetrieval",
```

```
"CreationDate": "2015-07-17T20:23:41.616Z",  
"StatusCode": "InProgress"  
}
```

L'ID del processo si trova nell'output di `aws glacier initiate-job` e `aws glacier list-jobs`. Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

- Per API i dettagli, vedere [DescribeJob](#) in AWS CLI Command Reference.

## describe-vault

Il seguente esempio di codice mostra come utilizzare `describe-vault`.

### AWS CLI

Il comando seguente recupera i dati relativi a un vault denominato `my-vault`:

```
aws glacier describe-vault --vault-name my-vault --account-id -
```

Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

- Per API i dettagli, vedere [DescribeVault](#) in AWS CLI Command Reference.

## get-data-retrieval-policy

Il seguente esempio di codice mostra come utilizzare `get-data-retrieval-policy`.

### AWS CLI

Il comando seguente ottiene la politica di recupero dei dati per l'account in uso:

```
aws glacier get-data-retrieval-policy --account-id -
```

Output:

```
{  
  "Policy": {  
    "Rules": [  
      {  
        "BytesPerHour": 10737418240,  

```



```

    "Strategy": "BytesPerHour"
  }
]
}
}

```

Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

- Per API i dettagli, vedere [GetDataRetrievalPolicy](#) in AWS CLI Command Reference.

## get-job-output

Il seguente esempio di codice mostra come utilizzare `get-job-output`.

### AWS CLI

Il comando seguente salva l'output di un processo di inventario del vault in un file nella directory corrente denominata `output.json`:

```

aws glacier get-job-output --account-id - --vault-name my-vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-R047Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqLNHS61ds04CnMW output.json

```

`job-id` è disponibile nell'output di `aws glacier list-jobs`. Nota che il nome del file di output è un argomento posizionale non preceduto dal nome di un'opzione. Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

Output:

```

{
  "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_mw6k76ZFGElWQX-
ybtRDvc2VkJPSDtfKmQrj0IRQLSGsNuDp-
AJV1u2ccmDSyDUMzWkbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw", "ArchiveDescription": "multipart
upload
test", "CreationDate": "2015-04-06T22:24:34Z", "Size": 3145728, "SHA256TreeHash": "9628195fcdcbcb
```

- Per API i dettagli, vedere [GetJobOutput](#) in AWS CLI Command Reference.

## get-vault-access-policy

Il seguente esempio di codice mostra come utilizzare `get-vault-access-policy`.

### AWS CLI

Per recuperare la politica di accesso di un vault

L'`get-vault-access-policy` esempio seguente recupera la politica di accesso per il vault specificato.

```
aws glacier get-vault-access-policy \
  --account-id 111122223333 \
  --vault-name example_vault
```

Output:

```
{
  "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\"}]}"
  }
}
```

- Per API i dettagli, vedere [GetVaultAccessPolicy](#) in AWS CLI Command Reference.

## get-vault-lock

Il seguente esempio di codice mostra come utilizzare `get-vault-lock`.

## AWS CLI

Per ottenere i dettagli di una serratura del vault

L'`get-vault-lock` seguente ha recuperato i dettagli sulla serratura per il vault specificato.

```
aws glacier get-vault-lock \
  --account-id - \
  --vault-name MyVaultName
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Get Vault Lock \(GETlock-policy\)](#) nella Amazon Glacier Developer Guide. API

- Per i API dettagli, consulta Command Reference. [GetVaultLock](#)AWS CLI

## `get-vault-notifications`

Il seguente esempio di codice mostra come utilizzare `get-vault-notifications`.

### AWS CLI

Il comando seguente ottiene una descrizione della configurazione della notifica per un vault denominato `my-vault`:

```
aws glacier get-vault-notifications --account-id - --vault-name my-vault
```

Output:

```
{
  "vaultNotificationConfig": {
    "Events": [
      "InventoryRetrievalCompleted",
      "ArchiveRetrievalCompleted"
    ],
    "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault"
  }
}
```

Se non è stata configurata alcuna notifica per il vault, viene restituito un errore. Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

- Per API i dettagli, vedere [GetVaultNotifications](#) in AWS CLI Command Reference.

## initiate-job

Il seguente esempio di codice mostra come utilizzare `initiate-job`.

### AWS CLI

Il comando seguente avvia un processo per ottenere un inventario del `my-vault` vault:

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-parameters
'{"Type": "inventory-retrieval"}
```

Output:

```
{
  "location": "/0123456789012/vaults/my-vault/jobs/
zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3R1oGduS7Eg-
R047Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW",
  "jobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3R1oGduS7Eg-
R047Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_Xq1NHS61ds04CnMW"
}
```

Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

Il comando seguente avvia un processo per recuperare un archivio dal vault: `my-vault`

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-parameters file://job-archive-retrieval.json
```

`job-archive-retrieval.json` è un JSON file nella cartella locale che specifica il tipo di lavoro, l'ID di archivio e alcuni parametri opzionali:

```
{
  "Type": "archive-retrieval",
  "ArchiveId": "kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGIEWQX-
ybtRDvc2VkJPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDumZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw",
  "Description": "Retrieve archive on 2015-07-17",
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-topic"
}
```

IDsGli archivi sono disponibili nell'output di `aws glacier upload-archive` e `aws glacier get-job-output`.

Output:

```
{
  "location": "/011685312445/vaults/mwunderl/jobs/l7IL5-
EkXyEY9Ws95fClzIbk205uLYaFdAY0i-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJ059-8N9aHWav",
  "jobId": "l7IL5-EkXy205uLYaFdAY0iEY9Ws95fClzIbk-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJ059-8N9aHWav"
}
```

Per dettagli sul formato dei parametri del lavoro, consulta [Initiate Job in API Amazon Glacier Reference](#).

- Per API i dettagli, consulta [Command `InitiateJob` Reference AWS CLI](#).

## initiate-multipart-upload

Il seguente esempio di codice mostra come utilizzare `initiate-multipart-upload`.

### AWS CLI

Il comando seguente avvia un caricamento in più parti su un archivio denominato `my-vault` con una dimensione della parte di 1 MiB (1024 x 1024 byte) per file:

```
aws glacier initiate-multipart-upload --account-id - --part-size 1048576 --vault-name my-vault --archive-description "multipart upload test"
```

Il parametro di descrizione dell'archivio è facoltativo. Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

Questo comando genera un ID di caricamento in caso di successo. Usa l'ID di caricamento quando carichi ogni parte del tuo archivio con `aws glacier upload-multipart-part`. Per ulteriori informazioni sui caricamenti multipart su Amazon Glacier utilizzando Amazon Glacier, AWS CLI consulta [Using Amazon Glacier](#) nella Guida per l'utente AWS CLI.

- Per i API dettagli, consulta [Command Reference](#). [InitiateMultipartUpload](#) AWS CLI

## initiate-vault-lock

Il seguente esempio di codice mostra come utilizzare `initiate-vault-lock`.

### AWS CLI

Per avviare il processo di blocco del vault

L'esempio seguente installa una politica di blocco del vault sull'archivio specificato e imposta lo stato di blocco del vault su `InProgress`. È necessario completare il processo chiamando `complete-vault-lock` entro 24 ore per impostare lo stato del blocco del vault. `Locked`

```
aws glacier initiate-vault-lock \
  --account-id - \
  --vault-name MyVaultName \
  --policy file://vault_lock_policy.json
```

Contenuto di `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"}}}]}}
```

L'output è l'ID del blocco del vault che è possibile utilizzare per completare il processo di blocco del vault.

```
{
  "lockId": "9QZgEXAMPLEPhvL6xEXAMPLE"
}
```

Per ulteriori informazioni, consulta [Initiate Vault Lock \(POSTlock-policy\)](#) nella Amazon Glacier Developer Guide. API

- Per i API dettagli, consulta Command Reference. [InitiateVaultLock](#)AWS CLI

## list-jobs

Il seguente esempio di codice mostra come utilizzare `list-jobs`.

### AWS CLI

Il comando seguente elenca i lavori in corso e quelli completati di recente per un vault denominato `my-vault`:

```
aws glacier list-jobs --account-id - --vault-name my-vault
```

Output:

```
{
  "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-
R047Yc6FxsdBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61ds04CnMW",
        "Action": "InventoryRetrieval",
        "CreationDate": "2015-07-17T20:23:41.616Z",
        "StatusCode": ""InProgress""
    }
]
}

```

Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

- Per API i dettagli, vedere [ListJobs](#) in AWS CLI Command Reference.

## list-multipart-uploads

Il seguente esempio di codice mostra come utilizzare `list-multipart-uploads`.

### AWS CLI

Il comando seguente mostra tutti i caricamenti multiparte in corso per un archivio denominato: `my-vault`

```
aws glacier list-multipart-uploads --account-id - --vault-name my-vault
```

Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

Per ulteriori informazioni sui caricamenti multiparte su Amazon Glacier utilizzando Amazon Glacier, AWS CLI consulta [Using Amazon Glacier](#) nella Guida per l'utente. AWS CLI



- Per i API dettagli, consulta Command Reference. [ListMultipartUploads](#) AWS CLI

## list-parts

Il seguente esempio di codice mostra come utilizzare `list-parts`.

### AWS CLI

Il comando seguente elenca le parti caricate per un caricamento in più parti su un archivio denominato: `my-vault`

```
aws glacier list-parts --account-id - --vault-name my-vault --upload-id "SYZi7qnL-YGqGwAm8Kn3BLP2E1NCvnB-5961R09CSaPmPwkYGH0qeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdxCs8ut-D"
```

Output:

```
{
  "MultipartUploadId": "SYZi7qnL-YGqGwAm8Kn3BLP2E1NCvnB-5961R09CSaPmPwkYGH0qeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdxCs8ut-D",
  "Parts": [
    {
      "RangeInBytes": "0-1048575",
      "SHA256TreeHash":
"e1f2a7cd6e047350f69b9f8cfa60fa60fe2f02802097a9a026360a7edc1f553"
    },
    {
      "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 richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

Per ulteriori informazioni sui caricamenti multiparte su Amazon Glacier utilizzando Amazon Glacier, AWS CLI consulta Using Amazon Glacier nella Guida per l'utente. AWS CLI

- Per i API dettagli, consulta Command Reference. [ListParts](#)AWS CLI

## **list-provisioned-capacity**

Il seguente esempio di codice mostra come utilizzare `list-provisioned-capacity`.

### AWS CLI

Per recuperare le unità di capacità assegnate

L'`list-provisioned-capacity`esempio seguente recupera i dettagli di tutte le unità di capacità assegnate per l'account specificato.

```
aws glacier list-provisioned-capacity \  
  --account-id 111122223333
```

Output:

```
{  
  "ProvisionedCapacityList": [  
    {  
      "CapacityId": "HpASAUvfRFiVDb0jMfEIcr8K",  
      "ExpirationDate": "2020-03-18T19:59:24.000Z",  
      "StartDate": "2020-02-18T19:59:24.912Z"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [ListProvisionedCapacity](#)in AWS CLI Command Reference.

## **list-tags-for-vault**

Il seguente esempio di codice mostra come utilizzare `list-tags-for-vault`.

### AWS CLI

Il comando seguente consente di elencare i tag applicati a un vault denominato `my-vault`:

```
aws glacier list-tags-for-vault --account-id - --vault-name my-vault
```

Output:

```
{
  "Tags": {
    "date": "july2015",
    "id": "1234"
  }
}
```

Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

- Per API i dettagli, vedere [ListTagsForVault](#) in AWS CLI Command Reference.

## list-vaults

Il seguente esempio di codice mostra come utilizzare `list-vaults`.

### AWS CLI

Il comando seguente consente di elencare i vault nell'account e nella regione predefiniti:

```
aws glacier list-vaults --account-id -
```

Output:

```
{
  "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 richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

- Per API i dettagli, vedere [ListVaults](#) in AWS CLI Command Reference.

## **purchase-provisioned-capacity**

Il seguente esempio di codice mostra come utilizzare `purchase-provisioned-capacity`.

### AWS CLI

Per acquistare un'unità di capacità assegnata

L'esempio seguente acquista un'unità di capacità assegnata.

```
aws glacier purchase-provisioned-capacity \  
  --account-id 111122223333
```

Output:

```
{  
  "capacityId": "HpASAuvfRFiVDb0jMfEIcr8K"  
}
```

- Per API i dettagli, vedere [PurchaseProvisionedCapacity](#) in AWS CLI Command Reference.

## **remove-tags-from-vault**

Il seguente esempio di codice mostra come utilizzare `remove-tags-from-vault`.

### AWS CLI

Il comando seguente rimuove un tag con la chiave `date` da un archivio denominato `my-vault`:

```
aws glacier remove-tags-from-vault --account-id - --vault-name my-vault --tag-  
keys date
```

Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

- Per API i dettagli, vedere [RemoveTagsFromVault](#) in AWS CLI Command Reference.

## **set-data-retrieval-policy**

Il seguente esempio di codice mostra come utilizzare `set-data-retrieval-policy`.

## AWS CLI

Il comando seguente configura una politica di recupero dei dati per l'account in uso:

```
aws glacier set-data-retrieval-policy --account-id - --policy file://data-retrieval-policy.json
```

`data-retrieval-policy.json` è un JSON file nella cartella corrente che specifica una politica di recupero dei dati:

```
{
  "Rules": [
    {
      "Strategy": "BytesPerHour",
      "BytesPerHour": 10737418240
    }
  ]
}
```

Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

Il comando seguente imposta la politica di recupero dei dati sull'utilizzo di inline: `FreeTier` JSON

```
aws glacier set-data-retrieval-policy --account-id - --policy '{"Rules": [{"Strategy": "FreeTier"}]}'
```

Per dettagli sul formato della policy, consulta la sezione [Set Data Retrieval Policy](#) in [Amazon API Glacier Reference](#).

- Per i API dettagli, consulta [SetDataRetrievalPolicy](#) Command Reference.AWS CLI

## **set-vault-access-policy**

Il seguente esempio di codice mostra come utilizzare `set-vault-access-policy`.

### AWS CLI

Per impostare la politica di accesso di un vault

L'`set-vault-access-policy` esempio seguente allega una politica di autorizzazione al vault specificato.

```
aws glacier set-vault-access-policy \
  --account-id 111122223333 \
  --vault-name example_vault
  --policy '{"Policy": "{\n"Version":\n"2012-10-17",\n"Statement":
  [\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"Effect":\n"Allow",\n"Principal":
  {\n"AWS":\n"arn:aws:iam:444455556666:root"},\n"Action":\n"glacier:UploadArchive",
  \n"Resource":\n"arn:aws:glacier:us-east-1:111122223333:vaults/example_vault"}]}'
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [SetVaultAccessPolicy](#) in AWS CLI Command Reference.

## set-vault-notifications

Il seguente esempio di codice mostra come utilizzare `set-vault-notifications`.

### AWS CLI

Il comando seguente configura SNS le notifiche per un archivio denominato: `my-vault`

```
aws glacier set-vault-notifications --account-id - --vault-name my-vault --vault-
notification-config file://notificationconfig.json
```

`notificationconfig.json` è un JSON file nella cartella corrente che specifica un SNS argomento e gli eventi da pubblicare:

```
{
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
  "Events": ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"]
}
```

Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

- Per API i dettagli, vedere [SetVaultNotifications](#) in AWS CLI Command Reference.

## upload-archive

Il seguente esempio di codice mostra come utilizzare `upload-archive`.

## AWS CLI

Il comando seguente carica un archivio nella cartella corrente denominata `archive.zip` in un archivio denominato `my-vault`:

```
aws glacier upload-archive --account-id - --vault-name my-vault --body archive.zip
```

Output:

```
{
  "archiveId": "kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGIEWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJV1lu2ccmDSyDUmZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw",
  "checksum": "969fb39823836d81f0cc028195fcdbcbbe76cdde932d4646fa7de5f21e18aa67",
  "location": "/0123456789012/vaults/my-vault/archives/
kKB7ymWJVpPSwhGP6ycS0Aekp9ZYe_--zM_mw6k76ZFGIEWQX-ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJV1lu2ccmDSyDUmZwKbwbpAdGATGDiB3hH00bjbGehXTcApVud_wyDw"
}
```

Amazon Glacier richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

Per recuperare un archivio caricato, avvia un processo di recupero con il comando `initiate-job` di `aws glacier`.

- Per API i dettagli, vedere [UploadArchive](#) in AWS CLI Command Reference.

## upload-multipart-part

Il seguente esempio di codice mostra come utilizzare `upload-multipart-part`.

### AWS CLI

Il comando seguente carica la prima parte da 1 MiB (1024 x 1024 byte) di un archivio:

```
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 richiede un argomento ID account durante l'esecuzione delle operazioni, ma puoi utilizzare un trattino per specificare l'account in uso.

Il parametro `body` indica il percorso di un file delle parti sul filesystem locale. Il parametro `range` accetta un intervallo di HTTP contenuto che indica i byte occupati dalla parte nell'archivio completo. L'ID di caricamento viene restituito dal comando `aws glacier initiate-multipart-upload` e può essere ottenuto anche utilizzando `aws glacier list-multipart-uploads`.

Per ulteriori informazioni sui caricamenti multiparte su Amazon Glacier utilizzando Amazon Glacier, AWS CLI consulta *Using Amazon Glacier* nella Guida per l'utente AWS CLI

- Per i API dettagli, consulta Command Reference. [UploadMultipartPart](#) AWS CLI

## Esempi di Secrets Manager che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Secrets Manager.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **batch-get-secret-value**

Il seguente esempio di codice mostra come utilizzare `batch-get-secret-value`.

#### AWS CLI

Esempio 1: recuperare il valore segreto per un gruppo di segreti elencati per nome

L'`batch-get-secret-value` esempio seguente ottiene il valore segreto `secrets` per tre segreti.

```
aws secretsmanager batch-get-secret-value \  
  --secret-id-list MySecret1 MySecret2 MySecret3
```



## Output:

```
{
  "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": []
}
```

```
}
```

Per ulteriori informazioni, consulta [Recuperare un gruppo di segreti in un batch nella Guida per l'utente di AWS Secrets Manager](#).

Esempio 2: Per recuperare il valore segreto per un gruppo di segreti selezionati dal filtro

L'batch-get-secret-value esempio seguente ottiene il valore segreto che i segreti presenti nell'account hanno MySecret nel nome. Il filtro per nome fa distinzione tra maiuscole e minuscole.

```
aws secretsmanager batch-get-secret-value \
  --filters Key="name",Values="MySecret"
```

Output:

```
{
  "SecretValues": [
    {
      "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret1-
a1b2c3",
      "Name": "MySecret1",
      "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEeaaaa",
      "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-EXAMPLEebbbb",
      "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-EXAMPLEecccc",
        "SecretString": "{\"username\":\"jie_liu\",\"password\":\"EXAMPLE-
PASSWORD\""}",
        "VersionStages": [
          "AWSCURRENT"
        ],
        "CreateDate": "1373477721.124"
      }
    ],
    "Errors": []
  }

```

Per ulteriori informazioni, consulta [Recuperare un gruppo di segreti in un batch nella Guida per l'utente di AWS Secrets Manager](#).

- Per API i dettagli, vedere [BatchGetSecretValue](#) in AWS CLI Command Reference.

## cancel-rotate-secret

Il seguente esempio di codice mostra come utilizzare `cancel-rotate-secret`.

### AWS CLI

Per disattivare la rotazione automatica di un segreto

L'esempio seguente disattiva la rotazione automatica di un segreto. Per riprendere la rotazione, chiamare `rotate-secret`.

```
aws secretsmanager cancel-rotate-secret \
  --secret-id MyTestSecret
```

Output:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret"
}
```

```
}
```

Per ulteriori informazioni, consulta [Ruotare un segreto](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [CancelRotateSecret](#) in AWS CLI Command Reference.

## create-secret

Il seguente esempio di codice mostra come utilizzare `create-secret`.

### AWS CLI

Esempio 1: creare un codice segreto dalle credenziali in un file JSON

L'esempio di `create-secret` seguente mostra come creare un segreto partendo dalle credenziali in un file. Per ulteriori informazioni, consultate [Caricamento AWS CLI dei parametri da un file](#) nella Guida per l'AWS CLI utente.

```
aws secretsmanager create-secret \  
  --name MyTestSecret \  
  --secret-string file://mycreds.json
```

Contenuto di `mycreds.json`.

```
{  
  "engine": "mysql",  
  "username": "saanvis",  
  "password": "EXAMPLE-PASSWORD",  
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",  
  "dbname": "myDatabase",  
  "port": "3306"  
}
```

Output:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",  
  "Name": "MyTestSecret",  
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Per ulteriori informazioni, consulta [Creare un segreto](#) nella Guida per l'utente di Secrets Manager.

### Esempio 2: creare un segreto

L'esempio di `create-secret` seguente mostra come creare un segreto con due coppie chiave-valore. Quando immetti i comandi in una shell dei comandi, c'è il rischio che la cronologia dei comandi sia accessibile o che le utilità abbiano accesso ai parametri dei comandi. Questo è un problema se il comando include il valore di un segreto. Per ulteriori informazioni, consulta [Mitigare i rischi dell'utilizzo di strumenti da riga di comando per archiviare segreti](#) nella Guida per l'utente di Secrets Manager.

```
aws secretsmanager create-secret \  
  --name MyTestSecret \  
  --description "My test secret created with the CLI." \  
  --secret-string "{\"user\": \"diegor\", \"password\": \"EXAMPLE-PASSWORD\"}"
```

Output:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",  
  "Name": "MyTestSecret",  
  "VersionId": "EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE"  
}
```

Per ulteriori informazioni, consulta [Creare un segreto](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [CreateSecret](#) in AWS CLI Command Reference.

## delete-resource-policy

Il seguente esempio di codice mostra come utilizzare `delete-resource-policy`.

### AWS CLI

Per eliminare la politica basata sulle risorse allegata a un segreto

L'esempio di `delete-resource-policy` seguente mostra come eliminare la policy basata su risorse collegata a un segreto.

```
aws secretsmanager delete-resource-policy \  

```

```
--secret-id MyTestSecret
```

Output:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Per ulteriori informazioni, consulta [Autenticazione e controllo degli accessi](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [DeleteResourcePolicy](#) in AWS CLI Command Reference.

## **delete-secret**

Il seguente esempio di codice mostra come utilizzare `delete-secret`.

### AWS CLI

Esempio 1: eliminare un segreto

L'esempio di `delete-secret` seguente mostra come eliminare un segreto. È possibile recuperare il segreto `restore-secret` entro la data e l'ora indicate nel campo di `DeletionDate` risposta. Per eliminare un segreto replicato in altre regioni, è necessario dapprima rimuovere le relative repliche con `remove-regions-from-replication`, quindi chiamare `delete-secret`.

```
aws secretsmanager delete-secret \
  --secret-id MyTestSecret \
  --recovery-window-in-days 7
```

Output:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret",
  "DeletionDate": 1524085349.095
}
```

```
}
```

Per ulteriori informazioni, consulta [Eliminare un segreto](#) nella Guida per l'utente di Secrets Manager.

Esempio 2: eliminare immediatamente un segreto

L'esempio di `delete-secret` seguente mostra come eliminare immediatamente il segreto senza un intervallo di recupero. Non è possibile recuperare questo segreto.

```
aws secretsmanager delete-secret \  
  --secret-id MyTestSecret \  
  --force-delete-without-recovery
```

Output:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",  
  "Name": "MyTestSecret",  
  "DeletionDate": 1508750180.309  
}
```

Per ulteriori informazioni, consulta [Eliminare un segreto](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [DeleteSecret](#) in AWS CLI Command Reference.

## **describe-secret**

Il seguente esempio di codice mostra come utilizzare `describe-secret`.

AWS CLI

Per recuperare i dettagli di un segreto

L'esempio seguente mostra i dettagli di un segreto.

```
aws secretsmanager describe-secret \  
  --secret-id MyTestSecret
```

## Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, [consulta Secret](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [DescribeSecret](#) in AWS CLI Command Reference.

## get-random-password

Il seguente esempio di codice mostra come utilizzare `get-random-password`.

### AWS CLI

Per generare una password casuale

L'`get-random-password` esempio seguente genera una password casuale lunga 20 caratteri che include almeno una lettera maiuscola, una lettera minuscola, un numero e una punteggiatura.

```
aws secretsmanager get-random-password \
  --require-each-included-type \
  --password-length 20
```

Output:

```
{
  "RandomPassword": "EXAMPLE-PASSWORD"
}
```

Per ulteriori informazioni, consulta [Creare e gestire segreti](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [GetRandomPassword](#) in AWS CLI Command Reference.

## get-resource-policy

Il seguente esempio di codice mostra come utilizzare `get-resource-policy`.

## AWS CLI

Per recuperare la politica basata sulle risorse allegata a un segreto

L'esempio di `get-resource-policy` seguente mostra come recuperare la policy basata su risorse collegata a un segreto.

```
aws secretsmanager get-resource-policy \  
  --secret-id MyTestSecret
```

Output:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret",  
  "ResourcePolicy": "{\n  \"Version\": \"2012-10-17\", \n  \"Statement\": [{\n    \"Effect\":  
    \"Allow\", \n    \"Principal\": {\n      \"AWS\": \"arn:aws:iam::123456789012:root\"\n    }, \n    \"Action\":  
    \"secretsmanager:GetSecretValue\", \n    \"Resource\": \"*\"\n  }]\n}"
```

Per ulteriori informazioni, consulta [Autenticazione e controllo degli accessi](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [GetResourcePolicy](#) in AWS CLI Command Reference.

## get-secret-value

Il seguente esempio di codice mostra come utilizzare `get-secret-value`.

### AWS CLI

Esempio 1: recuperare il valore segreto crittografato di un segreto

L'esempio di `get-secret-value` seguente mostra come recuperare il valore corrente del segreto.

```
aws secretsmanager get-secret-value \  
  --secret-id MyTestSecret
```

Output:

```
{
  "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
}
```

Per ulteriori informazioni, consulta [Recupera un segreto](#) nella Guida per l'utente di Secrets Manager.

Esempio 2: Per recuperare il valore segreto precedente

L'`get-secret-value` seguente ottiene il valore segreto precedente. :

```
aws secretsmanager get-secret-value \
  --secret-id MyTestSecret
  --version-stage AWSPREVIOUS
```

Output:

```
{
  "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
}
```

Per ulteriori informazioni, consulta [Recupera un segreto](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [GetSecretValue](#) in AWS CLI Command Reference.

## list-secret-version-ids

Il seguente esempio di codice mostra come utilizzare `list-secret-version-ids`.

### AWS CLI

Per elencare tutte le versioni segrete associate a un segreto

L'`list-secret-version-ids`esempio seguente ottiene un elenco di tutte le versioni di un segreto.

```
aws secretsmanager list-secret-version-ids \  
  --secret-id MyTestSecret
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, vedere [Versione](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [ListSecretVersionIds](#) in AWS CLI Command Reference.

## list-secrets

Il seguente esempio di codice mostra come utilizzare `list-secrets`.

### AWS CLI

Esempio 1: Per elencare i segreti del tuo account

L'esempio di `list-secrets` seguente mostra come ottenere un elenco dei segreti del proprio account.

```
aws secretsmanager list-secrets
```

Output:

```
{
  "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"
        ]
      }
    }
  ]
}
```

```
]
}
```

Per ulteriori informazioni, consulta [Find a secret](#) nella Guida per l'utente di Secrets Manager.

Esempio 2: per filtrare l'elenco dei segreti nel tuo account

L'`list-secrets` seguente ottiene un elenco dei segreti del tuo account contenuti Test nel nome. Il filtro per nome fa distinzione tra maiuscole e minuscole.

```
aws secretsmanager list-secrets \
  --filter Key="name",Values="Test"
```

Output:

```
{
  "SecretList": [
    {
      "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-a1b2c3",
      "Name": "MyTestSecret",
      "LastChangedDate": 1523477145.729,
      "SecretVersionsToStages": {
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
          "AWSCURRENT"
        ]
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta [Find a secret](#) nella Guida per l'utente di Secrets Manager.

Esempio 3: Per elencare i segreti presenti nel tuo account gestito da un altro servizio

L'`list-secrets` seguente restituisce i segreti del tuo account gestiti da AmazonRDS.

```
aws secretsmanager list-secrets \
  --filter Key="owning-service",Values="rds"
```

Output:

```

{
  "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-EXAMPLEaaaaa": [
          "AWSPREVIOUS"
        ],
        "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb": [
          "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"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Segreti gestiti da altri servizi](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [ListSecrets](#) in AWS CLI Command Reference.

## put-resource-policy

Il seguente esempio di codice mostra come utilizzare `put-resource-policy`.

### AWS CLI

Per aggiungere una politica basata sulle risorse a un segreto

L'esempio di `put-resource-policy` seguente mostra come aggiungere una policy di autorizzazioni a un segreto, verificando innanzitutto che la policy non fornisca un accesso ampio al segreto. La policy viene letta da un file. Per ulteriori informazioni, consulta [Caricamento AWS CLI dei parametri da un file nella Guida](#) per l'AWS CLI utente.

```
aws secretsmanager put-resource-policy \  
  --secret-id MyTestSecret \  
  --resource-policy file://mypolicy.json \  
  --block-public-policy
```

Contenuto di `mypolicy.json`.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::123456789012:role/MyRole"  
      },  
      "Action": "secretsmanager:GetSecretValue",  
      "Resource": "*"   
    }  
  ]  
}
```

Output:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",  
  "Name": "MyTestSecret"
```



```
}
```

Per ulteriori informazioni, consulta [Allegare una politica di autorizzazioni a un segreto](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [PutResourcePolicy](#) in AWS CLI Command Reference.

## put-secret-value

Il seguente esempio di codice mostra come utilizzare `put-secret-value`.

### AWS CLI

Esempio 1: per memorizzare un nuovo valore segreto in un segreto

L'`put-secret-value` seguente crea una nuova versione di un segreto con due coppie chiave-valore.

```
aws secretsmanager put-secret-value \  
  --secret-id MyTestSecret \  
  --secret-string "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}"
```

Output:

```
{  
  "ARN": "arn:aws:secretsmanager:us-  
west-2:123456789012:secret:MyTestSecret-1a2b3c",  
  "Name": "MyTestSecret",  
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "VersionStages": [  
    "AWSCURRENT"  
  ]  
}
```

Per ulteriori informazioni, consulta [Modificare un segreto](#) nella Guida per l'utente di Secrets Manager.

Esempio 2: Per memorizzare un nuovo valore segreto derivante dalle credenziali in un file JSON

L'esempio di `put-secret-value` seguente mostra come creare una nuova versione di un segreto partendo dalle credenziali in un file. Per ulteriori informazioni, consultate [Caricamento AWS CLI dei parametri da un file](#) nella Guida per l'AWS CLI utente.

```
aws secretsmanager put-secret-value \  
  --secret-id MyTestSecret \  
  --secret-string file://mycreds.json
```

Contenuto di `mycreds.json`.

```
{  
  "engine": "mysql",  
  "username": "saanvis",  
  "password": "EXAMPLE-PASSWORD",  
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",  
  "dbname": "myDatabase",  
  "port": "3306"  
}
```

Output:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",  
  "Name": "MyTestSecret",  
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
  "VersionStages": [  
    "AWSCURRENT"  
  ]  
}
```

Per ulteriori informazioni, consulta [Modificare un segreto](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [PutSecretValue](#) in AWS CLI Command Reference.

## **remove-regions-from-replication**

Il seguente esempio di codice mostra come utilizzare `remove-regions-from-replication`.

AWS CLI

Per eliminare una replica segreta

L'esempio di `remove-regions-from-replication` seguente mostra come eliminare un segreto di replica nella Regione `eu-west-3`. Per eliminare un segreto primario replicato in altre Regioni, è necessario dapprima eliminare le relative repliche e poi chiamare `delete-secret`.

```
aws secretsmanager remove-regions-from-replication \  
  --secret-id MyTestSecret \  
  --remove-replica-regions eu-west-3
```

Output:

```
{  
  "ARN": "arn:aws:secretsmanager:us-  
west-2:123456789012:secret:MyTestSecret-1a2b3c",  
  "ReplicationStatus": []  
}
```

Per ulteriori informazioni, consulta [Eliminare un segreto di replica](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [RemoveRegionsFromReplication](#) in AWS CLI Command Reference.

## **replicate-secret-to-regions**

Il seguente esempio di codice mostra come utilizzare `replicate-secret-to-regions`.

AWS CLI

Per replicare un segreto in un'altra regione

Nell'esempio `replicate-secret-to-regions` seguente, un segreto viene replicato nella Regione `eu-west-3`. La replica è crittografata con la chiave AWS gestita. `aws/secretsmanager`

```
aws secretsmanager replicate-secret-to-regions \  
  --secret-id MyTestSecret \  
  --add-replica-regions Region=eu-west-3
```

Output:

```
{
```

```
"ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-1a2b3c",
  "ReplicationStatus": [
    {
      "Region": "eu-west-3",
      "KmsKeyId": "alias/aws/secretsmanager",
      "Status": "InProgress"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Replicare un segreto in un'altra regione nella Guida per l'utente di Secrets Manager](#).

- Per API i dettagli, vedere [ReplicateSecretToRegions](#) in AWS CLI Command Reference.

## restore-secret

Il seguente esempio di codice mostra come utilizzare `restore-secret`.

### AWS CLI

Per ripristinare un segreto precedentemente eliminato

L'esempio di `restore-secret` seguente mostra il ripristino di un segreto per il quale in precedenza era stata pianificata l'eliminazione.

```
aws secretsmanager restore-secret \
  --secret-id MyTestSecret
```

Output:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Per ulteriori informazioni, consulta [Eliminare un segreto](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [RestoreSecret](#) in AWS CLI Command Reference.

## rotate-secret

Il seguente esempio di codice mostra come utilizzare `rotate-secret`.

### AWS CLI

Esempio 1: configurare e avviare la rotazione automatica di un segreto

L'esempio seguente configura e avvia la rotazione automatica di un segreto. Secrets Manager ruota il segreto una volta immediatamente e poi ogni otto ore in una finestra di due ore. L'output mostra `VersionId` la nuova versione segreta creata mediante rotazione.

```
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\"}"
```

Output:

```
{  
  "ARN": "aws:arn:secretsmanager:us-west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",  
  "Name": "MyTestDatabaseSecret",  
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Per ulteriori informazioni, consulta [Ruotare i segreti](#) nella Guida per l'utente di Secrets Manager.

Esempio 2: configurare e avviare la rotazione automatica su un intervallo di rotazione

L'esempio seguente configura e avvia la rotazione automatica per un segreto. Secrets Manager ruota il segreto una volta immediatamente e poi ogni 10 giorni. L'output mostra `VersionId` la nuova versione segreta creata mediante rotazione.

```
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)\"}"
```

**Output:**

```
{
  "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name": "MyTestDatabaseSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Per ulteriori informazioni, consulta [Ruotare i segreti](#) nella Guida per l'utente di Secrets Manager.

**Esempio 3: ruotare immediatamente un segreto**

L'esempio di `rotate-secret` seguente mostra come avviare una rotazione immediata. L'output mostra `VersionId` la nuova versione segreta creata mediante rotazione. Il segreto deve avere già la rotazione configurata.

```
aws secretsmanager rotate-secret \
  --secret-id MyTestDatabaseSecret
```

**Output:**

```
{
  "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
  "Name": "MyTestDatabaseSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Per ulteriori informazioni, consulta [Ruotare i segreti](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [RotateSecret](#) in AWS CLI Command Reference.

**stop-replication-to-replica**

Il seguente esempio di codice mostra come utilizzare `stop-replication-to-replica`.

**AWS CLI**

Promuovere una replica segreta a una principale

L'esempio di `stop-replication-to-replica` seguente mostra come rimuovere il collegamento tra un segreto di replica e quello primario. Il segreto di replica viene promosso a segreto primario nella Regione della replica. È necessario effettuare una chiamata `stop-replication-to-replica` dall'interno della Regione della replica.

```
aws secretsmanager stop-replication-to-replica \  
  --secret-id MyTestSecret
```

Output:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3"  
}
```

Per ulteriori informazioni, consulta [Promuovere un segreto di replica](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [StopReplicationToReplica](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

AWS CLI

Esempio 1: aggiungere un tag a un segreto

L'esempio di seguente mostra come collegare un tag con una sintassi abbreviata.

```
aws secretsmanager tag-resource \  
  --secret-id MyTestSecret \  
  --tags Key=FirstTag,Value=FirstValue
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagga i tuoi segreti](#) nella Guida per l'utente di Secrets Manager.

Esempio 2: Per aggiungere più tag a un segreto

L'esempio di `tag-resource` seguente mostra come collegare due tag chiave-valore a un segreto.

```
aws secretsmanager tag-resource \  
  --secret-id MyTestSecret \  
  --tags '[{"Key": "FirstTag", "Value": "FirstValue"}, {"Key": "SecondTag",  
  "Value": "SecondValue"}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tag secrets](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da un segreto

L'esempio di `untag-resource` seguente mostra come rimuovere due tag da un segreto. Per ogni tag, vengono rimossi sia la chiave che il valore.

```
aws secretsmanager untag-resource \  
  --secret-id MyTestSecret \  
  --tag-keys '[ "FirstTag", "SecondTag"]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tag secrets](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-secret-version-stage

Il seguente esempio di codice mostra come utilizzare `update-secret-version-stage`.

### AWS CLI

Esempio 1: ripristinare un segreto alla versione precedente



L'update-secret-version-stageesempio seguente sposta l'etichetta AWS CURRENT temporanea alla versione precedente di un segreto, che ripristina il segreto alla versione precedente. Per trovare l'ID della versione precedente, usa. list-secret-version-ids Per questo esempio, la versione con l'etichetta è a1b2c3d4-5678-90ab-cdef- e la versione con l' AWS CURRENTetichetta è a1b2c3d4-5678-90ab-cdef-. EXAMPLE11111 AWS PREVIOUS EXAMPLE22222 In questo esempio, si sposta l'etichetta dalla versione 11111 alla 22222. AWS CURRENT Poiché l' AWS CURRENTetichetta viene rimossa da una versione, sposta update-secret-version-stage automaticamente l' AWS PREVIOUSetichetta a quella versione (11111). L'effetto è che le AWS PREVIOUS versioni AWS CURRENT e vengono scambiate.

```
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
```

Output:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Per ulteriori informazioni, vedere [Versione](#) nella Guida per l'utente di Secrets Manager.

Esempio 2: per aggiungere un'etichetta temporanea allegata a una versione di un segreto

L'update-secret-version-stageesempio seguente aggiunge un'etichetta temporanea a una versione di un segreto. È possibile esaminare i risultati eseguendo list-secret-version-ids e visualizzando il campo di VersionStages risposta per la versione interessata.

```
aws secretsmanager update-secret-version-stage \
  --secret-id MyTestSecret \
  --version-stage STAGINGLABEL1 \
  --move-to-version-id EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE
```

Output:

```
{
```

```
"ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Per ulteriori informazioni, vedere [Versione](#) nella Guida per l'utente di Secrets Manager.

Esempio 3: Per eliminare un'etichetta temporanea associata a una versione di un segreto

L'update-secret-version-stageesempio seguente elimina un'etichetta temporanea associata a una versione di un segreto. È possibile esaminare i risultati eseguendo list-secret-version-ids e visualizzando il campo di VersionStages risposta per la versione interessata.

```
aws secretsmanager update-secret-version-stage \
  --secret-id MyTestSecret \
  --version-stage STAGINGLABEL1 \
  --remove-from-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
  "Name": "MyTestSecret"
}
```

Per ulteriori informazioni, vedere [Versione](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [UpdateSecretVersionStage](#) in AWS CLI Command Reference.

## update-secret

Il seguente esempio di codice mostra come utilizzareupdate-secret.

### AWS CLI

Esempio 1: Aggiornare la descrizione di un segreto

L'esempio di update-secret seguente mostra come aggiornare la descrizione di un segreto.

```
aws secretsmanager update-secret \
```

```
--secret-id MyTestSecret \  
--description "This is a new description for the secret."
```

Output:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret"  
}
```

Per ulteriori informazioni, consulta [Modificare un segreto](#) nella Guida per l'utente di Secrets Manager.

Esempio 2: Per aggiornare la chiave di crittografia associata a un segreto

L'update-secretesempio seguente aggiorna la KMS chiave utilizzata per crittografare il valore segreto. La KMS chiave deve trovarsi nella stessa area del segreto.

```
aws secretsmanager update-secret \  
  --secret-id MyTestSecret \  
  --kms-key-id arn:aws:kms:us-west-2:123456789012:key/EXAMPLE1-90ab-cdef-fedc-  
ba987EXAMPLE
```

Output:

```
{  
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-  
a1b2c3",  
  "Name": "MyTestSecret"  
}
```

Per ulteriori informazioni, consulta [Modificare un segreto](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [UpdateSecret](#) in AWS CLI Command Reference.

## **validate-resource-policy**

Il seguente esempio di codice mostra come utilizzare `validate-resource-policy`.

## AWS CLI

Per convalidare una politica delle risorse

L'`validate-resource-policy` seguente verifica che una politica delle risorse non conceda un ampio accesso a un segreto. La policy viene letta da un file su disco. Per ulteriori informazioni, consulta [Caricamento AWS CLI dei parametri da un file](#) nella Guida AWS CLI per l'utente.

```
aws secretsmanager validate-resource-policy \  
  --resource-policy file://mypolicy.json
```

Contenuto di `mypolicy.json`.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::123456789012:role/MyRole"  
      },  
      "Action": "secretsmanager:GetSecretValue",  
      "Resource": "*"   
    }  
  ]  
}
```

Output:

```
{  
  "PolicyValidationPassed": true,  
  "ValidationErrors": []  
}
```

Per ulteriori informazioni, consulta il [riferimento alle autorizzazioni per Secrets Manager](#) nella Guida per l'utente di Secrets Manager.

- Per API i dettagli, vedere [ValidateResourcePolicy](#) in AWS CLI Command Reference.

## Esempi di Security Hub che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Security Hub.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **accept-administrator-invitation**

Il seguente esempio di codice mostra come utilizzare `accept-administrator-invitation`.

AWS CLI

Accettare un invito da un account amministratore

L'esempio seguente accetta l'invito specificato dall'account amministratore specificato.

```
aws securityhub accept-invitation \  
  --administrator-id 123456789012 \  
  --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [AcceptAdministratorInvitation](#) in AWS CLI Command Reference.

## accept-invitation

Il seguente esempio di codice mostra come utilizzare `accept-invitation`.

### AWS CLI

Accettare un invito da un account amministratore

L'`accept-invitation` seguente accetta l'invito specificato dall'account amministratore specificato.

```
aws securityhub accept-invitation \  
  --master-id 123456789012 \  
  --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [AcceptInvitation](#) in AWS CLI Command Reference.

## batch-delete-automation-rules

Il seguente esempio di codice mostra come utilizzare `batch-delete-automation-rules`.

### AWS CLI

Per eliminare le regole di automazione

L'`batch-delete-automation-rules` seguente elimina la regola di automazione specificata. È possibile eliminare una o più regole con un solo comando. Solo l'account amministratore di Security Hub può eseguire questo comando.

```
aws securityhub batch-delete-automation-rules \  
  --automation-rules-arns ["arn:aws:securityhub:us-  
east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"]
```

Output:

```
{
```

```
"ProcessedAutomationRules": [  
  "arn:aws:securityhub:us-east-1:123456789012:automation-rule/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
],  
"UnprocessedAutomationRules": []  
}
```

Per ulteriori informazioni, vedere [Eliminazione delle regole di automazione](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [BatchDeleteAutomationRules](#) in AWS CLI Command Reference.

## batch-disable-standards

Il seguente esempio di codice mostra come utilizzare `batch-disable-standards`.

### AWS CLI

Per disabilitare uno standard

L'esempio seguente disabilita lo standard associato all'abbonamento ARN specificato.

```
aws securityhub batch-disable-standards \  
  --standards-subscription-arns "arn:aws:securityhub:us-  
west-1:123456789012:subscription/pci-dss/v/3.2.1"
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Disabilitazione o abilitazione di uno standard di sicurezza](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [BatchDisableStandards](#) in AWS CLI Command Reference.

## batch-enable-standards

Il seguente esempio di codice mostra come utilizzare `batch-enable-standards`.

### AWS CLI

Per abilitare uno standard

L'esempio seguente abilita lo PCI DSS standard per l'account richiedente.

```
aws securityhub batch-enable-standards \
  --standards-subscription-requests '{"StandardsArn":"arn:aws:securityhub:us-
west-1::standards/pci-dss/v/3.2.1"}'
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Disabilitazione o abilitazione di uno standard di sicurezza](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [BatchEnableStandards](#) in AWS CLI Command Reference.



## batch-get-automation-rules

Il seguente esempio di codice mostra come utilizzare `batch-get-automation-rules`.

### AWS CLI

Per ottenere dettagli sulle regole di automazione

L'esempio seguente ottiene i dettagli per la regola di automazione specificata. È possibile ottenere dettagli per una o più regole di automazione con un solo comando.

```
aws securityhub batch-get-automation-rules \
  --automation-rules-arns '["arn:aws:securityhub:us-
  east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"]'
```

Output:

```
{
  "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": []
}

```

Per ulteriori informazioni, vedere [Visualizzazione delle regole di automazione](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [BatchGetAutomationRules](#) in AWS CLI Command Reference.

## batch-get-configuration-policy-associations

Il seguente esempio di codice mostra come utilizzare `batch-get-configuration-policy-associations`.

### AWS CLI

Per ottenere i dettagli dell'associazione di configurazione per un batch di destinazioni

L'esempio seguente recupera i dettagli dell'associazione per gli obiettivi specificati. È possibile fornire l'account ID, l'unità organizzativa o l'ID radice per la destinazione.

```
aws securityhub batch-get-configuration-policy-associations \
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91k12m"}'
```

Output:

```
{
  "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
  "TargetId": "ou-6hi7-8j91k12m",
  "TargetType": "ORGANIZATIONAL_UNIT",
  "AssociationType": "APPLIED",
  "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
  "AssociationStatus": "SUCCESS",
  "AssociationStatusMessage": "Association applied successfully on this target."
}
```

Per ulteriori informazioni, vedere [Visualizzazione dei criteri di configurazione del Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [BatchGetConfigurationPolicyAssociations](#) in AWS CLI Command Reference.

## batch-get-security-controls

Il seguente esempio di codice mostra come utilizzare `batch-get-security-controls`.

### AWS CLI

Per ottenere i dettagli del controllo di sicurezza

L'batch-get-security-controls esempio seguente ottiene i dettagli per i controlli di sicurezza ACM .1 e IAM .1 nell' AWS account e AWS nella regione correnti.

```
aws securityhub batch-get-security-controls \  
--security-control-ids '["ACM.1", "IAM.1"]'
```

Output:

```
{  
  "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 \"*\" administrative  
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": {}
    }
]
}

```

Per ulteriori informazioni, vedere [Visualizzazione dei dettagli di un controllo nella Guida per l'utente del AWS Security Hub](#).

- Per API i dettagli, vedere [BatchGetSecurityControls](#) in AWS CLI Command Reference.

## batch-get-standards-control-associations

Il seguente esempio di codice mostra come utilizzare `batch-get-standards-control-associations`.

### AWS CLI

Per ottenere lo stato di attivazione di un controllo

L'esempio seguente identifica se i controlli specificati sono abilitati negli standard specificati.

```

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"}]'

```

Output:

```
{
```

```
"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"
    ]
  }
]
```

```
]
}
```

Per ulteriori informazioni, vedere [Abilitazione e disabilitazione dei controlli in standard specifici nella Guida per l'utente AWS di Security Hub](#).

- Per API i dettagli, vedere [BatchGetStandardsControlAssociations](#) in AWS CLI Command Reference.

## batch-import-findings

Il seguente esempio di codice mostra come utilizzare `batch-import-findings`.

### AWS CLI

Per aggiornare un risultato

L'`batch-import-findings`esempio seguente aggiorna un risultato.

```
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"
}]'
```

Output:

```

{
  "FailedCount": 0,
  "SuccessCount": 1,
  "FailedFindings": []
}
```

Per ulteriori informazioni, consulta [Utilizzare BatchImportFindings per creare e aggiornare i risultati](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [BatchImportFindings](#) in AWS CLI Command Reference.

## batch-update-automation-rules

Il seguente esempio di codice mostra come utilizzare `batch-update-automation-rules`.

### AWS CLI

Per aggiornare le regole di automazione

L'`batch-update-automation-rules` esempio seguente aggiorna la regola di automazione specificata. È possibile aggiornare una o più regole con un solo comando. Solo l'account amministratore di Security Hub può eseguire questo comando.

```

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" \
} \
]'

```

Output:

```

{
  "ProcessedAutomationRules": [
    "arn:aws:securityhub:us-east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
  ],
  "UnprocessedAutomationRules": []
}

```

Per ulteriori informazioni, consulta [Modifica delle regole di automazione](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [BatchUpdateAutomationRules](#) in AWS CLI Command Reference.

## batch-update-findings

Il seguente esempio di codice mostra come utilizzare `batch-update-findings`.

### AWS CLI

Esempio 1: aggiornare un risultato

L'batch-update-findingsesempio seguente aggiorna due risultati per aggiungere una nota, modificare l'etichetta di gravità e risolverlo.

```
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"}'
```

Output:

```
{
  "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": []
}
```

Per ulteriori informazioni, consulta [Utilizzare BatchUpdateFindings per aggiornare un risultato](#) nella Guida per l'utente AWS di Security Hub.

Esempio 2: Per aggiornare un risultato utilizzando la sintassi abbreviata

L'batch-update-findingsesempio seguente aggiorna due risultati per aggiungere una nota, modificare l'etichetta di gravità e risolverlo utilizzando una sintassi abbreviata.

```
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"

```

Output:

```

{
  "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": []
}

```

Per ulteriori informazioni, consulta [Utilizzare BatchUpdateFindings per aggiornare un risultato](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [BatchUpdateFindings](#) in AWS CLI Command Reference.

## batch-update-standards-control-associations

Il seguente esempio di codice mostra come utilizzare `batch-update-standards-control-associations`.

AWS CLI

Per aggiornare lo stato di abilitazione di un controllo negli standard abilitati

L'batch-update-standards-control-associationsempio seguente disabilita CloudTrail .1 negli standard specificati.

```
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"}]'
```

Questo comando non produce alcun output se ha esito positivo.

Per ulteriori informazioni, vedere [Abilitazione e disabilitazione dei controlli in standard specifici](#) e [Abilitazione e disabilitazione dei controlli in tutti gli standard nella Guida](#) per l'utente di AWS Security Hub.

- Per API i dettagli, vedere [BatchUpdateStandardsControlAssociations](#) in AWS CLI Command Reference.

## create-action-target

Il seguente esempio di codice mostra come utilizzare create-action-target.

### AWS CLI

Per creare un'azione personalizzata

L'create-action-targetesempio seguente crea un'azione personalizzata. Fornisce il nome, la descrizione e l'identificatore dell'azione.

```
aws securityhub create-action-target \  
  --name "Send to remediation" \  
  --description "Action to send the finding for remediation tracking" \  
  --id "Remediation"
```

Output:

```
{
```

```
"ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/Remediation"
}
```

Per ulteriori informazioni, vedere [Creazione di un'azione personalizzata e associazione a una regola CloudWatch Events](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [CreateActionTarget](#) in AWS CLI Command Reference.

## create-automation-rule

Il seguente esempio di codice mostra come utilizzare `create-automation-rule`.

### AWS CLI

Per creare una regola di automazione

L'esempio seguente crea una regola di automazione nell'account corrente e AWS nella regione. Security Hub filtra i risultati in base ai criteri specificati e applica le azioni ai risultati corrispondenti. Solo l'account amministratore di Security Hub può eseguire questo comando.

```
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"
```

Output:

```
{  
  "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE111111"  
}
```

Per ulteriori informazioni, vedere [Creazione di regole di automazione](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [CreateAutomationRule](#) in AWS CLI Command Reference.

## create-configuration-policy

Il seguente esempio di codice mostra come utilizzare `create-configuration-policy`.

AWS CLI

Per creare una politica di configurazione

L'`create-configuration-policy` esempio seguente crea una politica di configurazione con le impostazioni specificate.

```
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"}'
```

Output:

```

{
  "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
                }
              }
            }
          }
        ]
      }
    }
  }
}

```

Per ulteriori informazioni, vedere [Creazione e associazione dei criteri di configurazione del Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [CreateConfigurationPolicy](#) in AWS CLI Command Reference.

## create-finding-aggregator

Il seguente esempio di codice mostra come utilizzare `create-finding-aggregator`.

### AWS CLI

Per abilitare l'aggregazione dei risultati

L'esempio seguente configura l'aggregazione dei risultati. Viene eseguita dagli Stati Uniti orientali (Virginia), che designano gli Stati Uniti orientali (Virginia) come regione di aggregazione. Indica di collegare solo regioni specifiche e di non collegare automaticamente nuove regioni. Seleziona Stati Uniti occidentali (California settentrionale) e Stati Uniti occidentali (Oregon) come regioni collegate.

```
aws securityhub create-finding-aggregator \  
  --region us-east-1 \  
  --region-linking-mode SPECIFIED_REGIONS \  
  --regions us-west-1,us-west-2
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Enabling finding aggregation](#) nella AWS Security Hub User Guide.

- Per API i dettagli, vedere [CreateFindingAggregator](#) in AWS CLI Command Reference.

## create-insight

Il seguente esempio di codice mostra come utilizzare `create-insight`.

### AWS CLI

Per creare una visione personalizzata



L'`create-insight`esempio seguente crea un'analisi personalizzata denominata `Critical role finding` che restituisce risultati critici correlati ai AWS ruoli.

```
aws securityhub create-insight \  
  --filters '{"ResourceType": [{ "Comparison": "EQUALS", "Value": "AwsIamRole"}],  
  "SeverityLabel": [{"Comparison": "EQUALS", "Value": "CRITICAL"}]}' \  
  --group-by-attribute "ResourceId" \  
  --name "Critical role findings"
```

Output:

```
{  
  "InsightArn": "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
}
```

Per ulteriori informazioni, consulta [Gestire informazioni personalizzate](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [CreateInsight](#)in AWS CLI Command Reference.

## `create-members`

Il seguente esempio di codice mostra come utilizzare `create-members`.

AWS CLI

Per aggiungere account come account membri

L'`create-members`esempio seguente aggiunge due account come account membro all'account amministratore richiedente.

```
aws securityhub create-members \  
  --account-details '[{"AccountId": "123456789111"}, {"AccountId":  
  "123456789222"}]'
```

Output:

```
{  
  "UnprocessedAccounts": []
```

```
}
```

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [CreateMembers](#) in AWS CLI Command Reference.

## decline-invitations

Il seguente esempio di codice mostra come utilizzare `decline-invitations`.

### AWS CLI

Per rifiutare un invito a diventare membro dell'account

L'`decline-invitations` seguente rifiuta un invito a diventare un account membro dell'account amministratore specificato. L'account membro è l'account richiedente.

```
aws securityhub decline-invitations \
  --account-ids "123456789012"
```

Output:

```
{
  "UnprocessedAccounts": []
}
```

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DeclineInvitations](#) in AWS CLI Command Reference.

## delete-action-target

Il seguente esempio di codice mostra come utilizzare `delete-action-target`.

### AWS CLI

Per eliminare un'azione personalizzata

L'`delete-action-target` seguente elimina l'azione personalizzata identificata dall'azione specificata ARN.

```
aws securityhub delete-action-target \  
  --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/  
Remediation"
```

Output:

```
{  
  "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/  
Remediation"  
}
```

Per ulteriori informazioni, vedere [Creazione di un'azione personalizzata e associazione a una regola CloudWatch Events](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DeleteActionTarget](#) in AWS CLI Command Reference.

## delete-configuration-policy

Il seguente esempio di codice mostra come utilizzare `delete-configuration-policy`.

AWS CLI

Per eliminare una politica di configurazione

L'`delete-configuration-policy` esempio seguente elimina la politica di configurazione specificata.

```
aws securityhub delete-configuration-policy \  
  --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-  
policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Eliminazione e dissociazione dei criteri di configurazione di Security Hub](#) nella Guida per l'utente di AWS Security Hub.

- Per API i dettagli, vedere [DeleteConfigurationPolicy](#) in AWS CLI Command Reference.

## delete-finding-aggregator

Il seguente esempio di codice mostra come utilizzare `delete-finding-aggregator`.

## AWS CLI

Per smettere di cercare l'aggregazione

L'`delete-finding-aggregatoresempio` seguente interrompe la ricerca dell'aggregazione. Viene eseguito dagli Stati Uniti orientali (Virginia), che è la regione di aggregazione.

```
aws securityhub delete-finding-aggregator \  
  --region us-east-1 \  
  --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-agg  
regator/123e4567-e89b-12d3-a456-426652340000
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Stopping finding aggregation](#) nella AWS Security Hub User Guide.

- Per API i dettagli, vedere [DeleteFindingAggregator](#) in AWS CLI Command Reference.

## `delete-insight`

Il seguente esempio di codice mostra come utilizzare `delete-insight`.

### AWS CLI

Per eliminare un insight personalizzato

L'`delete-insightesempio` seguente elimina l'analisi personalizzata con quella specificataARN.

```
aws securityhub delete-insight \  
  --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Output:

```
{  
  "InsightArn": "arn:aws:securityhub:eu-  
central-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-  
EXAMPLE11111"  
}
```

Per ulteriori informazioni, consulta [Gestire informazioni personalizzate](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DeleteInsight](#) in AWS CLI Command Reference.

## delete-invitations

Il seguente esempio di codice mostra come utilizzare `delete-invitations`.

### AWS CLI

Per eliminare un invito a diventare membro dell'account

L'`delete-invitation`sempio seguente elimina un invito a diventare un account membro per l'account amministratore specificato. L'account membro è l'account richiedente.

```
aws securityhub delete-invitations \  
  --account-ids "123456789012"
```

Output:

```
{  
  "UnprocessedAccounts": []  
}
```

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DeleteInvitations](#) in AWS CLI Command Reference.

## delete-members

Il seguente esempio di codice mostra come utilizzare `delete-members`.

### AWS CLI

Per eliminare gli account dei membri

L'`delete-members`sempio seguente elimina gli account membro specificati dall'account amministratore richiedente.

```
aws securityhub delete-members \  
  --account-ids "123456789012"
```

```
--account-ids "123456789111" "123456789222"
```

Output:

```
{
  "UnprocessedAccounts": []
}
```

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DeleteMembers](#) in AWS CLI Command Reference.

## describe-action-targets

Il seguente esempio di codice mostra come utilizzare `describe-action-targets`.

AWS CLI

Per recuperare i dettagli sulle azioni personalizzate

L'esempio seguente recupera le informazioni sull'azione personalizzata identificata dall'azione specificata. ARN

```
aws securityhub describe-action-targets \
  --action-target-arns "arn:aws:securityhub:us-west-1:123456789012:action/custom/
  Remediation"
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Creazione di un'azione personalizzata e associazione a una regola CloudWatch Events](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DescribeActionTargets](#) in AWS CLI Command Reference.

## describe-hub

Il seguente esempio di codice mostra come utilizzare `describe-hub`.

### AWS CLI

Per ottenere informazioni su una risorsa dell'hub

L'esempio seguente restituisce la data di sottoscrizione per la risorsa hub specificata. La risorsa hub è identificata dalla sua ARN.

```
aws securityhub describe-hub \  
  --hub-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

Output:

```
{  
  "HubArn": "arn:aws:securityhub:us-west-1:123456789012:hub/default",  
  "SubscribedAt": "2019-11-19T23:15:10.046Z"  
}
```

Per ulteriori informazioni, vedere [AWS::SecurityHub: :Hub](#) nella Guida per l'AWS CloudFormation utente.

- Per API i dettagli, vedere [DescribeHub](#) in AWS CLI Command Reference.

## describe-organization-configuration

Il seguente esempio di codice mostra come utilizzare `describe-organization-configuration`.

### AWS CLI

Per visualizzare come è configurato Security Hub per un'organizzazione

L'esempio seguente restituisce informazioni sul modo in cui un'organizzazione è configurata in Security Hub. In questo esempio, l'organizzazione

utilizza la configurazione centrale. Solo l'account amministratore di Security Hub può eseguire questo comando.

```
aws securityhub describe-organization-configuration
```

Output:

```
{
  "AutoEnable": false,
  "MemberAccountLimitReached": false,
  "AutoEnableStandards": "NONE",
  "OrganizationConfiguration": {
    "ConfigurationType": "LOCAL",
    "Status": "ENABLED",
    "StatusMessage": "Central configuration has been enabled successfully"
  }
}
```

Per ulteriori informazioni, consulta [Managing accounts with AWS Organizations](#) nella AWS Security Hub User Guide.

- Per API i dettagli, vedere [DescribeOrganizationConfiguration](#) in AWS CLI Command Reference.

## describe-products

Il seguente esempio di codice mostra come utilizzare `describe-products`.

AWS CLI

Per restituire informazioni sulle integrazioni di prodotti disponibili

L'`describe-product` esempio seguente restituisce le integrazioni di prodotti disponibili una alla volta.

```
aws securityhub describe-products \
  --max-results 1
```

Output:

```
{
```



```

    "NextToken": "U2FsdGVkX18vvP10qb7RD1rWRWVFBJI46M0IAb+nZmRjM15NoRi2gm13sdQEn30/
    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\\\"}}}}]"
      }
    ]
  }

```

Per ulteriori informazioni, consulta [Gestire le integrazioni dei prodotti](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DescribeProducts](#) in AWS CLI Command Reference.

## describe-standards-controls

Il seguente esempio di codice mostra come utilizzare `describe-standards-controls`.

### AWS CLI

Per richiedere l'elenco dei controlli in uno standard abilitato

L'esempio seguente richiede l'elenco dei controlli nella sottoscrizione allo PCI DSS standard dell'account richiedente. La richiesta restituisce due controlli alla volta.

```
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
```

Output:

```
{
  "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+eNkPoZHVl11ip5HUYQPWSWZGmfctmJiHL8JoKEsCDuaKayiPDyLK
+LiTkShveo0dvfxXCk0BaGhohIXhsIedN+LSjQV/
17kfCfJcq4PziNC1N9xe9aq2pjlLVZnznTfSImrodT5bRNHe4fELCQq/z+5ka
+5Lzmc11axcwTd5lKgQyQqmUVoeriHZhyIiBgWKf7oNYdBVG80EortVWvSkoUTt
+B2ThcnC7l43kI0UNx1kZ6sc64AsW"
}

```

Per ulteriori informazioni, consulta [Visualizzazione dei dettagli per i controlli](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DescribeStandardsControls](#) in AWS CLI Command Reference.

## describe-standards

Il seguente esempio di codice mostra come utilizzare `describe-standards`.

### AWS CLI

Per restituire un elenco di standard disponibili

L'`describe-standards`esempio seguente restituisce l'elenco degli standard disponibili.

```
aws securityhub describe-standards
```

**Output:**

```
{
  "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
    }
  ]
}
```

Per ulteriori informazioni, consulta [gli standard di sicurezza in AWS Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DescribeStandards](#) in AWS CLI Command Reference.

## disable-import-findings-for-product

Il seguente esempio di codice mostra come utilizzare `disable-import-findings-for-product`.

### AWS CLI

Per non ricevere più risultati dall'integrazione di un prodotto

L'`disable-import-findings-for-product` esempio seguente disabilita il flusso di risultati per l'abbonamento specificato a un'integrazione di prodotto.

```
aws securityhub disable-import-findings-for-product \  
  --product-subscription-arn "arn:aws:securityhub:us-west-1:123456789012:product-  
  subscription/crowdstrike/crowdstrike-falcon"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Gestire le integrazioni dei prodotti](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DisableImportFindingsForProduct](#) in AWS CLI Command Reference.

## disable-organization-admin-account

Il seguente esempio di codice mostra come utilizzare `disable-organization-admin-account`.

### AWS CLI

Per rimuovere un account amministratore di Security Hub

L'`disable-organization-admin-account` esempio seguente revoca l'assegnazione dell'account specificato come account amministratore di Security Hub per Organizations. AWS

```
aws securityhub disable-organization-admin-account \  
  --admin-account-id 777788889999
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Designazione di un account amministratore di Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DisableOrganizationAdminAccount](#) in AWS CLI Command Reference.

## **disable-security-hub**

Il seguente esempio di codice mostra come utilizzare `disable-security-hub`.

### AWS CLI

Per disabilitare AWS Security Hub

L'esempio seguente disattiva AWS Security Hub per l'account richiedente.

```
aws securityhub disable-security-hub
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Disabling AWS Security Hub nella Security Hub User Guide](#).AWS

- Per API i dettagli, vedere [DisableSecurityHub](#) in AWS CLI Command Reference.

## **disassociate-from-administrator-account**

Il seguente esempio di codice mostra come utilizzare `disassociate-from-administrator-account`.

### AWS CLI

Dissociarsi da un account amministratore

L'esempio seguente dissocia l'account richiedente dal relativo account amministratore corrente.

```
aws securityhub disassociate-from-administrator-account
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DisassociateFromAdministratorAccount](#) in AWS CLI Command Reference.

## **disassociate-from-master-account**

Il seguente esempio di codice mostra come utilizzare `disassociate-from-master-account`.

### AWS CLI

Dissociarsi da un account amministratore

L'`disassociate-from-master-account` esempio seguente dissocia l'account richiedente dal relativo account amministratore corrente.

```
aws securityhub disassociate-from-master-account
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DisassociateFromMasterAccount](#) in AWS CLI Command Reference.

## **disassociate-members**

Il seguente esempio di codice mostra come utilizzare `disassociate-members`.

### AWS CLI

Per dissociare gli account dei membri

L'`disassociate-members` esempio seguente dissocia gli account membro specificati dall'account amministratore richiedente.

```
aws securityhub disassociate-members \  
  --account-ids "123456789111" "123456789222"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [DisassociateMembers](#) in AWS CLI Command Reference.

## **enable-import-findings-for-product**

Il seguente esempio di codice mostra come utilizzare `enable-import-findings-for-product`.

### AWS CLI

Per iniziare a ricevere i risultati dell'integrazione di un prodotto

L'esempio seguente abilita il flusso di risultati derivanti dall'integrazione del prodotto specificata.

```
aws securityhub enable-import-findings-for-product \  
  --product-arn "arn:aws:securityhub:us-east-1:123456789333:product/crowdstrike/  
crowdstrike-falcon"
```

Output:

```
{  
  "ProductSubscriptionArn": "arn:aws:securityhub:us-east-1:123456789012:product-  
subscription/crowdstrike/crowdstrike-falcon"  
}
```

Per ulteriori informazioni, consulta [Gestire le integrazioni dei prodotti](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [EnableImportFindingsForProduct](#) in AWS CLI Command Reference.

## **enable-organization-admin-account**

Il seguente esempio di codice mostra come utilizzare `enable-organization-admin-account`.

### AWS CLI

Per designare un account dell'organizzazione come account amministratore di Security Hub

L'esempio seguente designa l'account specificato come account amministratore di Security Hub.

```
aws securityhub enable-organization-admin-account \  
  --admin-account-id 777788889999
```



Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Designazione di un account amministratore di Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [EnableOrganizationAdminAccount](#) in AWS CLI Command Reference.

## **enable-security-hub**

Il seguente esempio di codice mostra come utilizzare `enable-security-hub`.

### AWS CLI

Per abilitare AWS Security Hub

L'esempio seguente abilita AWS Security Hub per l'account richiedente. Configura Security Hub per abilitare gli standard predefiniti. Per la risorsa hub, assegna il valore Security al tag. Department

```
aws securityhub enable-security-hub \  
  --enable-default-standards \  
  --tags '{"Department": "Security"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Enabling Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [EnableSecurityHub](#) in AWS CLI Command Reference.

## **get-administrator-account**

Il seguente esempio di codice mostra come utilizzare `get-administrator-account`.

### AWS CLI

Per recuperare informazioni su un account amministratore

L'esempio seguente recupera le informazioni sull'account amministratore per l'account richiedente.

```
aws securityhub get-administrator-account
```

**Output:**

```
{
  "Master": {
    "AccountId": "123456789012",
    "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
    "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
    "MemberStatus": "ASSOCIATED"
  }
}
```

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [GetAdministratorAccount](#) in AWS CLI Command Reference.

**get-configuration-policy-association**

Il seguente esempio di codice mostra come utilizzare `get-configuration-policy-association`.

**AWS CLI**

Per ottenere i dettagli dell'associazione di configurazione per un obiettivo

L'`get-configuration-policy-association` esempio seguente recupera i dettagli dell'associazione per l'obiettivo specificato. È possibile fornire un ID account, un ID dell'unità organizzativa o l'ID radice per la destinazione.

```
aws securityhub get-configuration-policy-association \
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91k12m"}'
```

**Output:**

```
{
  "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
  "TargetId": "ou-6hi7-8j91k12m",
  "TargetType": "ORGANIZATIONAL_UNIT",
  "AssociationType": "APPLIED",
  "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
  "AssociationStatus": "SUCCESS",
  "AssociationStatusMessage": "Association applied successfully on this target."
}
```

```
}
```

Per ulteriori informazioni, vedere [Visualizzazione dei criteri di configurazione del Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [GetConfigurationPolicyAssociation](#) in AWS CLI Command Reference.

## get-configuration-policy

Il seguente esempio di codice mostra come utilizzare `get-configuration-policy`.

### AWS CLI

Per visualizzare i dettagli della politica di configurazione

L'`get-configuration-policy` esempio seguente recupera i dettagli sulla politica di configurazione specificata.

```
aws securityhub get-configuration-policy \  
  --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Output:

```
{  
  "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
        }
      }
    }
  }
]
}
}
```

Per ulteriori informazioni, vedere [Visualizzazione dei criteri di configurazione del Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [GetConfigurationPolicy](#) in AWS CLI Command Reference.

## get-enabled-standards

Il seguente esempio di codice mostra come utilizzare `get-enabled-standards`.

### AWS CLI

Per recuperare informazioni su uno standard abilitato

L' `get-enabled-standards` esempio seguente recupera informazioni sullo PCI DSS standard.

```
aws securityhub get-enabled-standards \
  --standards-subscription-arn "arn:aws:securityhub:us-
  west-1:123456789012:subscription/pci-dss/v/3.2.1"
```

Output:

```
{
```

```

    "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [gli standard di sicurezza in AWS Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [GetEnabledStandards](#) in AWS CLI Command Reference.

## get-finding-aggregator

Il seguente esempio di codice mostra come utilizzare `get-finding-aggregator`.

### AWS CLI

Per recuperare la configurazione corrente dell'aggregazione dei risultati

L'`get-finding-aggregator` seguente recupera la configurazione corrente dell'aggregazione dei risultati.

```

aws securityhub get-finding-aggregator \
  --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-
aggregator/123e4567-e89b-12d3-a456-426652340000

```

Output:

```

{
  "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"
}

```

Per ulteriori informazioni, vedere [Visualizzazione della configurazione corrente dell'aggregazione dei risultati](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [GetFindingAggregator](#) in AWS CLI Command Reference.

## get-finding-history

Il seguente esempio di codice mostra come utilizzare `get-finding-history`.

### AWS CLI

Per iniziare a trovare la cronologia

L'`get-finding-history` esempio seguente recupera gli ultimi 90 giorni della cronologia del risultato specificato. In questo esempio, i risultati sono limitati a due record della cronologia dei risultati.

```
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"
```

Output:

```
{
  "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": []
}
]
}

```

Per ulteriori informazioni, consulta la sezione [Ricerca della cronologia](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [GetFindingHistory](#) in AWS CLI Command Reference.

## get-findings

Il seguente esempio di codice mostra come utilizzare `get-findings`.

### AWS CLI

Esempio 1: restituire i risultati generati per uno standard specifico

L'`get-findings` seguente restituisce i risultati per lo PCI DSS standard.

```
aws securityhub get-findings \
  --filters '{"GeneratorId":[{"Value": "pci-dss", "Comparison": "PREFIX"}]}' \
  --max-items 1
```

Output:

```
{
  "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": "eyJ0ZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxZjQ=="
}

```

Esempio 2: per restituire risultati di gravità critica con uno stato del flusso di lavoro pari a NOTIFIED

L'get-findingsesempio seguente restituisce risultati con un valore dell'etichetta di gravità pari a CRITICAL e uno stato del flusso di lavoro pari a. NOTIFIED I risultati vengono ordinati in ordine decrescente in base al valore di Confidence.

```

aws securityhub get-findings \
  --filters '{"SeverityLabel":[{"Value":
"CRITICAL", "Comparison": "EQUALS"}], "WorkflowStatus":
[{"Value": "NOTIFIED", "Comparison": "EQUALS"}]}' \
  --sort-criteria '{ "Field": "Confidence", "SortOrder": "desc"}' \
  --max-items 1

```

Output:

```

{
  "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"
  }
]
```

Per ulteriori informazioni, consulta [Filtraggio e raggruppamento dei risultati](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [GetFindings](#) in AWS CLI Command Reference.

## get-insight-results

Il seguente esempio di codice mostra come utilizzare `get-insight-results`.

### AWS CLI

Per recuperare i risultati per un approfondimento

L'`get-insight-result`sempio seguente restituisce l'elenco dei risultati di analisi per l'analisi con il valore specificato ARN.

```
aws securityhub get-insight-results \  
  --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Output:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

Per ulteriori informazioni, consulta [Visualizzazione e azione in base ai risultati e ai risultati di Insight](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [GetInsightResults](#) in AWS CLI Command Reference.

## get-insights

Il seguente esempio di codice mostra come utilizzare `get-insights`.

### AWS CLI

Per recuperare i dettagli di un'intuizione

L'`get-insights` esempio seguente recupera i dettagli di configurazione per l'analisi con quanto specificato. ARN

```
aws securityhub get-insights \
  --insight-arns "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
  custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Output:

```
{
  "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"
      }
    ]
  }
```

Per ulteriori informazioni, consulta [Insights in AWS Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [GetInsights](#) in AWS CLI Command Reference.

## get-invitations-count

Il seguente esempio di codice mostra come utilizzare `get-invitations-count`.

### AWS CLI

Per recuperare il numero di inviti che non sono stati accettati

L'esempio seguente recupera il numero di inviti che l'account richiedente ha rifiutato o a cui non ha risposto.

```
aws securityhub get-invitations-count
```

Output:

```
{
  "InvitationsCount": 3
}
```

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [GetInvitationsCount](#) in AWS CLI Command Reference.

## get-master-account

Il seguente esempio di codice mostra come utilizzare `get-master-account`.

### AWS CLI

Per recuperare informazioni su un account amministratore

L'get-master-accountesempio seguente recupera le informazioni sull'account amministratore per l'account richiedente.

```
aws securityhub get-master-account
```

Output:

```
{
  "Master": {
    "AccountId": "123456789012",
    "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
    "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
    "MemberStatus": "ASSOCIATED"
  }
}
```

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [GetMasterAccount](#) in AWS CLI Command Reference.

## get-members

Il seguente esempio di codice mostra come utilizzareget-members.

AWS CLI

Per recuperare informazioni sugli account dei membri selezionati

L'get-membersesempio seguente recupera le informazioni sugli account membro specificati.

```
aws securityhub get-members \
  --account-ids "444455556666" "777788889999"
```

Output:

```
{
  "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": [ ]
}

```

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [GetMembers](#) in AWS CLI Command Reference.

## get-security-control-definition

Il seguente esempio di codice mostra come utilizzare `get-security-control-definition`.

### AWS CLI

Per ottenere i dettagli della definizione del controllo di sicurezza

L'esempio seguente recupera i dettagli della definizione per un controllo di sicurezza Security Hub. I dettagli includono il titolo del controllo, la descrizione, la disponibilità della regione, i parametri e altre informazioni.

```
aws securityhub get-security-control-definition \
  --security-control-id ACM.1
```

Output:

```
{
  "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
          }
        }
      }
    }
  }
}

```

Per ulteriori informazioni, vedere [Parametri di controllo personalizzati](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [GetSecurityControlDefinition](#) in AWS CLI Command Reference.

## invite-members

Il seguente esempio di codice mostra come utilizzare `invite-members`.

### AWS CLI

Per inviare inviti agli account dei membri

L'`invite-members`esempio seguente invia gli inviti agli account dei membri specificati.

```
aws securityhub invite-members \  
  --account-ids "123456789111" "123456789222"
```

Output:

```
{  
  "UnprocessedAccounts": []  
}
```

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [InviteMembers](#) in AWS CLI Command Reference.

## list-automation-rules

Il seguente esempio di codice mostra come utilizzare `list-automation-rules`.

AWS CLI

Per visualizzare un elenco di regole di automazione

L'`list-automation-rules` esempio seguente elenca le regole di automazione per un AWS account. Solo l'account amministratore di Security Hub può eseguire questo comando.

```
aws securityhub list-automation-rules \  
  --max-results 3 \  
  --next-token NULL
```

Output:

```
{  
  "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"
  }
]
}

```

Per ulteriori informazioni, vedere [Visualizzazione delle regole di automazione](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [ListAutomationRules](#) in AWS CLI Command Reference.

## list-configuration-policies

Il seguente esempio di codice mostra come utilizzare `list-configuration-policies`.

### AWS CLI

Per elencare i riepiloghi delle politiche di configurazione

L'`list-configuration-policies` seguente elenca un riepilogo delle politiche di configurazione per l'organizzazione.

```
aws securityhub list-configuration-policies \  
  --max-items 3
```

Output:

```
{  
  "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  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Visualizzazione dei criteri di configurazione del Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [ListConfigurationPolicies](#) in AWS CLI Command Reference.

## list-configuration-policy-associations

Il seguente esempio di codice mostra come utilizzare `list-configuration-policy-associations`.

### AWS CLI

Per elencare le associazioni di configurazione

L'esempio seguente elenca un riepilogo delle associazioni di configurazione per l'organizzazione. La risposta include associazioni con politiche di configurazione e comportamenti autogestiti.

```
aws securityhub list-configuration-policy-associations \
  --association-type "APPLIED" \
  --max-items 4
```

### Output:

```
{
  "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-6hi7-8j91k12m",
        "TargetType": "ORGANIZATIONAL_UNIT",
        "AssociationType": "APPLIED",
        "UpdatedAt": "2023-09-26T21:13:01.816000+00:00",
        "AssociationStatus": "SUCCESS",
        "AssociationStatusMessage": "Association applied successfully on this
target."
    },
    {
        "ConfigurationPolicyId": "SELF_MANAGED_SECURITY_HUB",
        "TargetId": "111122223333",
        "TargetType": "ACCOUNT",
        "AssociationType": "APPLIED",
        "UpdatedAt": "2023-11-28T22:01:26.409000+00:00",
        "AssociationStatus": "SUCCESS"
    }
}

```

Per ulteriori informazioni, vedere [Visualizzazione dei criteri di configurazione del Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [ListConfigurationPolicyAssociations](#) in AWS CLI Command Reference.

## list-enabled-products-for-import

Il seguente esempio di codice mostra come utilizzare `list-enabled-products-for-import`.

### AWS CLI

Per restituire l'elenco delle integrazioni di prodotti abilitate

L'`list-enabled-products-for-import` esempio seguente restituisce l'elenco degli abbonamenti ARNS per le integrazioni di prodotti attualmente abilitate.

```
aws securityhub list-enabled-products-for-import
```

Output:

```

{
  "ProductSubscriptions": [ "arn:aws:securityhub:us-west-1:123456789012:product-
subscription/crowdstrike/crowdstrike-falcon", "arn:aws:securityhub:us-
west-1:123456789012:product-subscription/aws/securityhub" ]
}

```

```
}
```

Per ulteriori informazioni, consulta [Gestire le integrazioni dei prodotti](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [ListEnabledProductsForImport](#) in AWS CLI Command Reference.

## **list-finding-aggregators**

Il seguente esempio di codice mostra come utilizzare `list-finding-aggregators`.

### AWS CLI

Per elencare i widget disponibili

L'`list-finding-aggregators` seguente restituisce la configurazione ARN di aggregazione dei risultati.

```
aws securityhub list-finding-aggregators
```

Output:

```
{
  "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-
  aggregator/123e4567-e89b-12d3-a456-426652340000"
}
```

Per ulteriori informazioni, vedere [Visualizzazione della configurazione corrente dell'aggregazione dei risultati](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [ListFindingAggregators](#) in AWS CLI Command Reference.

## **list-invitations**

Il seguente esempio di codice mostra come utilizzare `list-invitations`.

### AWS CLI

Per visualizzare un elenco di inviti

L'`list-invitations` seguente recupera l'elenco degli inviti inviati all'account richiedente.



```
aws securityhub list-invitations
```

Output:

```
{
  "Invitations": [
    {
      "AccountId": "123456789012",
      "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
      "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
      "MemberStatus": "ASSOCIATED"
    }
  ],
}
```

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [ListInvitations](#) in AWS CLI Command Reference.

## list-members

Il seguente esempio di codice mostra come utilizzare `list-members`.

### AWS CLI

Per recuperare un elenco degli account dei membri

L'`list-members` esempio seguente restituisce l'elenco degli account membro per l'account amministratore richiedente.

```
aws securityhub list-members
```

Output:

```
{
  "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
  }
],
}
```

Per ulteriori informazioni, vedere [Gestione degli account amministratore e membro](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [ListMembers](#) in AWS CLI Command Reference.

## list-organization-admin-accounts

Il seguente esempio di codice mostra come utilizzare `list-organization-admin-accounts`.

### AWS CLI

Per elencare gli account amministrativi designati per il Security Hub

L'`list-organization-admin-accounts` seguente elenca gli account amministratore di Security Hub per un'organizzazione.

```
aws securityhub list-organization-admin-accounts
```

Output:

```
{
  AdminAccounts": [
    { "AccountId": "777788889999" },
    { "Status": "ENABLED" }
  ]
}
```

Per ulteriori informazioni, vedere [Designazione di un account amministratore di Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [ListOrganizationAdminAccounts](#) in AWS CLI Command Reference.

## list-security-control-definitions

Il seguente esempio di codice mostra come utilizzare `list-security-control-definitions`.

### AWS CLI

Esempio 1: Per elencare tutti i controlli di sicurezza disponibili

L'`list-security-control-definitions` seguente elenca i controlli di sicurezza disponibili in tutti gli standard di Security Hub. Questo esempio limita i risultati a tre controlli.

```
aws securityhub list-security-control-definitions \
  --max-items 3
```

Output:

```
{
  "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=="
}

```

Per ulteriori informazioni, vedere [Visualizzazione dei dettagli di uno standard nella Guida per l'utente del AWS Security Hub](#).

Esempio 2: per elencare i controlli di sicurezza disponibili per uno standard specifico

L'`list-security-control-definitions` seguente elenca i controlli di sicurezza disponibili per CIS AWS Foundations Benchmark v1.4.0. Questo esempio limita i risultati a tre controlli.

```
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
```

## Output:

```
{  
  "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=="
  }

```

Per ulteriori informazioni, vedere [Visualizzazione dei dettagli di uno standard nella Guida per l'utente del AWS Security Hub](#).

- Per API i dettagli, vedere [ListSecurityControlDefinitions](#) in AWS CLI Command Reference.

## list-standards-control-associations

Il seguente esempio di codice mostra come utilizzare `list-standards-control-associations`.

### AWS CLI

Per ottenere lo stato di attivazione di un controllo in ogni standard abilitato

L'`list-standards-control-associations` esempio seguente elenca lo stato di abilitazione di `CloudTrail .1` in ogni standard abilitato.

```

aws securityhub list-standards-control-associations \
  --security-control-id CloudTrail.1

```

### Output:

```

{
  "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)."
  }
]
}

```

Per ulteriori informazioni, vedere [Abilitazione e disabilitazione dei controlli in standard specifici nella Guida per l'utente AWS di Security Hub](#).

- Per API i dettagli, vedere [ListStandardsControlAssociations](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.



## AWS CLI

Per recuperare i tag assegnati a una risorsa

L'`list-tags-for-resource`esempio seguente restituisce i tag assegnati alla risorsa hub specificata.

```
aws securityhub list-tags-for-resource \  
  --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

Output:

```
{  
  "Tags": {  
    "Department" : "Operations",  
    "Area" : "USMidwest"  
  }  
}
```

Per ulteriori informazioni, vedere [AWS::SecurityHub: :Hub](#) nella Guida per l'AWS CloudFormation utente.

- Per API i dettagli, vedere [ListTagsForResource](#)in AWS CLI Command Reference.

## start-configuration-policy-association

Il seguente esempio di codice mostra come utilizzare `start-configuration-policy-association`.

### AWS CLI

Esempio 1: associare una politica di configurazione

L'`start-configuration-policy-association`esempio seguente associa la politica di configurazione specificata all'unità organizzativa specificata. Una configurazione può essere associata a un account di destinazione, a un'unità organizzativa o alla radice.

```
aws securityhub start-configuration-policy-association \  
  --configuration-policy-identifier "arn:aws:securityhub:eu-  
  central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333" \  
  --resource-arn "arn:aws:securityhub:eu-central-1:123456789012:hub/default"
```

```
--target '{"OrganizationalUnitId": "ou-6hi7-8j91k12m"}'
```

Output:

```
{
  "ConfigurationPolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
  "TargetId": "ou-6hi7-8j91k12m",
  "TargetType": "ORGANIZATIONAL_UNIT",
  "AssociationType": "APPLIED",
  "UpdatedAt": "2023-11-29T17:40:52.468000+00:00",
  "AssociationStatus": "PENDING"
}
```

Per ulteriori informazioni, vedere [Creazione e associazione dei criteri di configurazione del Security Hub](#) nella Guida per l'utente AWS di Security Hub.

Esempio 2: associare una configurazione autogestita

L'`start-configuration-policy-association` seguente associa una configurazione autogestita all'account specificato.

```
aws securityhub start-configuration-policy-association \
  --configuration-policy-identifier "SELF_MANAGED_SECURITY_HUB" \
  --target '{"OrganizationalUnitId": "123456789012"}'
```

Output:

```
{
  "ConfigurationPolicyId": "SELF_MANAGED_SECURITY_HUB",
  "TargetId": "123456789012",
  "TargetType": "ACCOUNT",
  "AssociationType": "APPLIED",
  "UpdatedAt": "2023-11-29T17:40:52.468000+00:00",
  "AssociationStatus": "PENDING"
}
```

Per ulteriori informazioni, vedere [Creazione e associazione dei criteri di configurazione del Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [StartConfigurationPolicyAssociation](#) in AWS CLI Command Reference.

## start-configuration-policy-disassociation

Il seguente esempio di codice mostra come utilizzare `start-configuration-policy-disassociation`.

### AWS CLI

Esempio 1: dissociare una politica di configurazione

L'esempio seguente `start-configuration-policy-disassociation` dissocia una politica di configurazione dall'unità organizzativa specificata. Una configurazione può essere dissociata da un account di destinazione, da un'unità organizzativa o dalla radice.

```
aws securityhub start-configuration-policy-disassociation \  
  --configuration-policy-identifier "arn:aws:securityhub:eu-  
central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333" \  
  --target '{"OrganizationalUnitId": "ou-6hi7-8j91k12m"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Dissociazione di una configurazione dagli account e OUs](#) nella Guida per l'utente AWS di Security Hub.

Esempio 2: dissociare una configurazione autogestita

L'esempio seguente `start-configuration-policy-disassociation` dissocia una configurazione autogestita dall'account specificato.

```
aws securityhub start-configuration-policy-disassociation \  
  --configuration-policy-identifier "SELF_MANAGED_SECURITY_HUB" \  
  --target '{"AccountId": "123456789012"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Dissociazione di una configurazione dagli account e OUs](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [StartConfigurationPolicyDisassociation](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per assegnare un tag a una risorsa

L'esempio seguente assegna i valori per i tag `Reparto` e `Area` alla risorsa `hub` specificata.

```
aws securityhub tag-resource \  
  --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \  
  --tags '{"Department":"Operations", "Area":"USMidwest"}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [AWS::SecurityHub: :Hub nella Guida](#) per l'AWS CloudFormation utente.

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un valore di tag da una risorsa

L'esempio seguente rimuove il tag `Department` dalla risorsa `hub` specificata.

```
aws securityhub untag-resource \  
  --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \  
  --tag-keys "Department"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [AWS::SecurityHub: :Hub](#) nella Guida per l'AWS CloudFormation utente.

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-action-target

Il seguente esempio di codice mostra come utilizzare `update-action-target`.

### AWS CLI

Per aggiornare un'azione personalizzata

L'`update-action-target`esempio seguente aggiorna il nome dell'azione personalizzata identificata dall'azione specificataARN.

```
aws securityhub update-action-target \  
  --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/  
Remediation" \  
  --name "Send to remediation"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Creazione di un'azione personalizzata e associazione a una regola CloudWatch Events](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [UpdateActionTarget](#)in AWS CLI Command Reference.

## update-configuration-policy

Il seguente esempio di codice mostra come utilizzare `update-configuration-policy`.

### AWS CLI

Per aggiornare una politica di configurazione

L'`update-configuration-policy`esempio seguente aggiorna una politica di configurazione esistente per utilizzare le impostazioni specificate.

```
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"
```

## Output:

```
{
  "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
                }
              }
            }
          }
        ]
      }
    }
  }
}
```

```
    }  
  }  
}
```

Per ulteriori informazioni, vedere [Aggiornamento dei criteri di configurazione del Security Hub](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [UpdateConfigurationPolicy](#) in AWS CLI Command Reference.

## update-finding-aggregator

Il seguente esempio di codice mostra come utilizzare `update-finding-aggregator`.

### AWS CLI

Per aggiornare la configurazione corrente dell'aggregazione dei risultati

L'`update-finding-aggregator` seguente modifica la configurazione dell'aggregazione dei risultati in base al collegamento da regioni selezionate. Viene gestito dagli Stati Uniti orientali (Virginia), che è la regione di aggregazione. Seleziona Stati Uniti occidentali (California settentrionale) e Stati Uniti occidentali (Oregon) come regioni collegate.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Aggiornamento della configurazione di aggregazione](#) dei risultati nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [UpdateFindingAggregator](#) in AWS CLI Command Reference.

## update-insight

Il seguente esempio di codice mostra come utilizzare `update-insight`.

## AWS CLI

Esempio 1: Per modificare il filtro per un'analisi personalizzata

L'update-insightesempio seguente modifica i filtri per un'analisi personalizzata. L'analisi aggiornata cerca risultati con una severità elevata correlati ai AWS ruoli.

```
aws securityhub update-insight \  
  --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/  
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111" \  
  --filters '{"ResourceType": [{ "Comparison": "EQUALS", "Value": "AwsIamRole"}],  
"SeverityLabel": [{"Comparison": "EQUALS", "Value": "HIGH"}]}' \  
  --name "High severity role findings"
```

Esempio 2: modificare l'attributo di raggruppamento per un'analisi personalizzata

L'update-insightesempio seguente modifica l'attributo di raggruppamento per l'analisi personalizzata con il valore specificato. ARN Il nuovo attributo di raggruppamento è l'ID della risorsa.

```
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"
```

Output:

```
{  
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Gestire informazioni personalizzate](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [UpdateInsight](#) in AWS CLI Command Reference.

## update-organization-configuration

Il seguente esempio di codice mostra come utilizzare `update-organization-configuration`.

### AWS CLI

Per aggiornare il modo in cui Security Hub è configurato per un'organizzazione

L'`update-organization-configuration` esempio seguente specifica che Security Hub deve utilizzare la configurazione centrale per configurare un'organizzazione. Dopo aver eseguito questo comando, l'amministratore delegato del Security Hub può creare e gestire i criteri di configurazione per configurare l'organizzazione. L'amministratore delegato può utilizzare questo comando anche per passare dalla configurazione centrale a quella locale. Se la configurazione locale è il tipo di configurazione, l'amministratore delegato può scegliere se abilitare automaticamente Security Hub e gli standard di sicurezza predefiniti nei nuovi account dell'organizzazione.

```

aws securityhub update-organization-configuration \
  --no-auto-enable \
  --organization-configuration '{"ConfigurationType": "CENTRAL"}'

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing accounts with AWS Organizations](#) nella AWS Security Hub User Guide.

- Per API i dettagli, vedere [UpdateOrganizationConfiguration](#) in AWS CLI Command Reference.

## update-security-control

Il seguente esempio di codice mostra come utilizzare `update-security-control`.

### AWS CLI

Per aggiornare le proprietà dei controlli di sicurezza

L'esempio seguente specifica valori personalizzati per un parametro di controllo di sicurezza Security Hub.

```
aws securityhub update-security-control \  
  --security-control-id ACM.1 \  
  --parameters '{"daysToExpiration": {"ValueType": "CUSTOM", "Value": {"Integer":  
15}}}' \  
  --last-update-reason "Internal compliance requirement"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Parametri di controllo personalizzati](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [UpdateSecurityControl](#) in AWS CLI Command Reference.

## update-security-hub-configuration

Il seguente esempio di codice mostra come utilizzare `update-security-hub-configuration`.

### AWS CLI

Per aggiornare la configurazione del Security Hub

L'esempio seguente configura Security Hub per abilitare automaticamente nuovi controlli per gli standard abilitati.

```
aws securityhub update-security-hub-configuration \  
  --auto-enable-controls
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Abilitazione automatica dei nuovi controlli](#) nella Guida per l'utente AWS di Security Hub.

- Per API i dettagli, vedere [UpdateSecurityHubConfiguration](#) in AWS CLI Command Reference.

## update-standards-control

Il seguente esempio di codice mostra come utilizzare `update-standards-control`.

### AWS CLI

Esempio 1: disattivare un controllo

L'`update-standards-control` seguente disabilita il PCI. AutoScaling1. Controllo.

```
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"
```

Questo comando non produce alcun output.

Esempio 2: abilitare un controllo

L'`update-standards-control` seguente abilita il PCI. AutoScaling1. Controllo.

```
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"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Disabilitazione e abilitazione dei singoli controlli](#) nella Guida per l'utente del AWS Security Hub.

- Per API i dettagli, vedere [UpdateStandardsControl](#) in AWS CLI Command Reference.

## Esempi di utilizzo di Security Lake AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con Security Lake.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-aws-logsource**

Il seguente esempio di codice mostra come utilizzare `create-aws-logsource`.

AWS CLI

Per aggiungere un Amazon Web Service con supporto nativo come sorgente Amazon Security Lake

L'`create-aws-logsource` esempio seguente aggiunge VPC Flow Logs come fonte Security Lake negli account e nelle regioni designati.

```
aws securitylake create-aws-log-source \  
  --sources '[{"regions": ["us-east-1"], "accounts": ["123456789012"],  
  "sourceName": "SH_FINDINGS", "sourceVersion": "2.0"}]'
```

Output:

```
{  
  "failed": [  
    "123456789012"  
  ]  
}
```

Per ulteriori informazioni, consulta [Aggiungere un AWS servizio come fonte](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [CreateAwsLogsource AWS CLI Command Reference](#).

## create-custom-logsource

Il seguente esempio di codice mostra come utilizzare `create-custom-logsource`.

### AWS CLI

Per aggiungere una fonte personalizzata come sorgente Amazon Security Lake

L'esempio seguente aggiunge una fonte personalizzata come origine Security Lake nell'account del provider di log designato e nella regione designata.

```
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/E1WG1ZNPXT0D4"}, "providerIdentity": {"principal": "029189416600", "externalId": "123456789012"}}' --region "us-east-1"
```

Output:

```
{
  "customLogSource": {
    "attributes": {
      "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/E1WG1ZNPXT0D4",
      "databaseArn": "arn:aws:glue:eu-west-2:123456789012:database/E1WG1ZNPXT0D4",
      "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/E1WG1ZNPXT0D4"
    },
    "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"
  }
}
```

```
}
```

Per ulteriori informazioni, consulta [Aggiungere una fonte personalizzata](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [CreateCustomLogsource AWS CLI Command Reference](#).

## **create-data-lake-exception-subscription**

Il seguente esempio di codice mostra come utilizzare `create-data-lake-exception-subscription`.

### AWS CLI

Per inviare notifiche relative alle eccezioni di Security Lake

L'esempio seguente invia notifiche relative alle eccezioni di Security Lake all'account specificato tramite SMS la consegna. Il messaggio di eccezione rimane valido per il periodo di tempo specificato.

```
aws securitylake create-data-lake-exception-subscription \  
  --notification-endpoint "123456789012" \  
  --exception-time-to-live 30 \  
  --subscription-protocol "sms"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Risoluzione dei problemi di Amazon Security Lake](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [CreateDataLakeExceptionSubscription AWS CLI Command Reference](#).

## **create-data-lake-organization-configuration**

Il seguente esempio di codice mostra come utilizzare `create-data-lake-organization-configuration`.

### AWS CLI

Per configurare Security Lake nei nuovi account dell'organizzazione

L'`create-data-lake-organization-configuration` seguente abilita Security Lake e la raccolta degli eventi di origine specificati e registra i nuovi account dell'organizzazione.

```
aws securitylake create-data-lake-organization-configuration \
  --auto-enable-new-account '[{"region":"us-east-1","sources":
  [{"sourceName":"SH_FINDINGS","sourceVersion":"1.0"}]}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing multiple account with AWS Organizations](#) nella Amazon Security Lake User Guide.

- Per API i dettagli, consulta [CreateDataLakeOrganizationConfiguration AWS CLI Command Reference](#).

## create-data-lake

Il seguente esempio di codice mostra come utilizzare `create-data-lake`.

### AWS CLI

Esempio 1: per configurare il data lake in più regioni

L'`create-data-lake` seguente abilita Amazon Security Lake in più AWS regioni e configura il tuo 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"
```

Output:

```
{
  "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-
gnev76s8z7bzby8oi3uiaysbr8v2ml",
    "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-
cehuifz15rwmhm6m62h7zhvtseogr9",
        "updateStatus": {
            "exception": {},
            "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
            "status": "INITIALIZED"
        }
    }
]
}

```

Per ulteriori informazioni, consulta la sezione [Guida introduttiva ad Amazon Security Lake](#) nella Guida per l'utente di Amazon Security Lake.

Esempio 2: configurare il data lake in una singola regione

L'`create-data-lake` seguente abilita Amazon Security Lake in una singola AWS regione e configura il tuo 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"

```

Output:

```
{
  "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"
      }
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione [Guida introduttiva ad Amazon Security Lake](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [CreateDataLake AWS CLI Command Reference](#).

## create-subscriber-data-access

Il seguente esempio di codice mostra come utilizzare `create-subscriber-data-access`.

### AWS CLI

Per creare un abbonato con accesso ai dati

L'esempio seguente crea un abbonato in Security Lake con accesso ai dati nella AWS regione corrente per l'identità di sottoscrittore specificata per un'origine. AWS

```
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}"'
```

Output:

```
{  
  "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"
}
}

```

Per ulteriori informazioni, consulta [Creare un abbonato con accesso ai dati](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [CreateSubscriberDataAccess AWS CLI Command Reference](#).

## create-subscriber-notification

Il seguente esempio di codice mostra come utilizzare `create-subscriber-notification`.

### AWS CLI

Per creare una notifica per gli abbonati

L'esempio seguente mostra come specificare la notifica del sottoscrittore per creare una notifica quando nuovi dati vengono scritti nel data lake.

```

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"}}'

```

Output:

```

{
  "subscriberEndpoint": [
    "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"
  ]
}

```

Per ulteriori informazioni, consulta [Gestione degli abbonati](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [CreateSubscriberNotification AWS CLI Command Reference](#).

## create-subscriber-query-access

Il seguente esempio di codice mostra come utilizzare `create-subscriber-query-access`.

### AWS CLI

Per creare un abbonato con accesso tramite query

L'`create-subscriber` seguente crea un sottoscrittore in Security Lake con accesso alle query nella AWS regione corrente per l'identità di sottoscrittore specificata.

```
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"}'
```

Output:

```
{  
  "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"
}
}
```

Per ulteriori informazioni, consulta [Creare un abbonato con accesso tramite query](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [CreateSubscriberQueryAccess AWS CLI Command Reference](#).

## delete-aws-logsources

Il seguente esempio di codice mostra come utilizzare `delete-aws-logsources`.

### AWS CLI

Per rimuovere un servizio supportato nativamente AWS .

L'esempio seguente elimina VPC Flow Logs come origine Security Lake negli account e nelle regioni designati.

```
aws securitylake delete-aws-log-source \
  --sources '[{"regions": ["us-east-1"], "accounts": ["123456789012"],
  "sourceName": "SH_FINDINGS", "sourceVersion": "2.0"}]'
```

Output:

```
{
  "failed": [
    "123456789012"
  ]
}
```

Per ulteriori informazioni, consulta [Rimuovere un AWS servizio come fonte](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [DeleteAwsLogsource AWS CLI Command Reference](#).

## delete-custom-logsource

Il seguente esempio di codice mostra come utilizzare `delete-custom-logsource`.

### AWS CLI

Per rimuovere una fonte personalizzata.

L'esempio seguente elimina un'origine personalizzata nell'account del provider di log designato nella regione designata.

```
aws securitylake delete-custom-log-source \  
  --source-name "CustomSourceName"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Eliminazione di un'origine personalizzata](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [DeleteCustomLogsource AWS CLI Command Reference](#).

## delete-data-lake-organization-configuration

Il seguente esempio di codice mostra come utilizzare `delete-data-lake-organization-configuration`.

### AWS CLI

Per interrompere la raccolta automatica dei sorgenti negli account dei membri

L'esempio seguente interrompe la raccolta automatica dei risultati del AWS Security Hub dai nuovi account membri che entrano a far parte dell'organizzazione. Solo l'amministratore delegato di Security Lake può eseguire questo comando. Impedisce ai nuovi account membro di fornire automaticamente dati al data lake.

```
aws securitylake delete-data-lake-organization-configuration \  
  --auto-enable-new-account '[{"region": "us-east-1", "sources":  
  [{"sourceName": "SH_FINDINGS"}]}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing multiple account with AWS Organizations](#) nella Amazon Security Lake User Guide.

- Per API i dettagli, consulta [DeleteDataLakeOrganizationConfiguration AWS CLI Command Reference](#).

## **delete-data-lake**

Il seguente esempio di codice mostra come utilizzare `delete-data-lake`.

### AWS CLI

Per disabilitare il tuo data lake

L'esempio seguente disattiva il data lake nelle AWS regioni specificate. Nelle regioni specificate, le fonti non forniscono più dati al data lake. Per una distribuzione di Security Lake che utilizza AWS Organizations, solo l'amministratore delegato di Security Lake dell'organizzazione può disabilitare Security Lake per gli account dell'organizzazione.

```
aws securitylake delete-data-lake \  
  --regions "ap-northeast-1" "eu-central-1"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Disabilitazione di Amazon Security Lake](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [DeleteDataLake AWS CLI Command Reference](#).

## **delete-subscriber-notification**

Il seguente esempio di codice mostra come utilizzare `delete-subscriber-notification`.

### AWS CLI

Per eliminare una notifica di sottoscrizione

L'esempio seguente mostra come eliminare la notifica di sottoscrizione per un abbonato specifico di Security Lake.

```
aws securitylake delete-subscriber-notification \  
  --subscriber-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```



Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Gestione degli abbonati](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [DeleteSubscriberNotification AWS CLI Command Reference](#).

## **delete-subscriber**

Il seguente esempio di codice mostra come utilizzare `delete-subscriber`.

### AWS CLI

Per eliminare un abbonato

L'esempio seguente mostra come rimuovere un abbonato se non si desidera più che un abbonato utilizzi i dati da Security Lake.

```
aws securitylake delete-subscriber \  
--subscriber-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Gestione degli abbonati](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [DeleteSubscriber AWS CLI Command Reference](#).

## **get-data-lake-exception-subscription**

Il seguente esempio di codice mostra come utilizzare `get-data-lake-exception-subscription`.

### AWS CLI

Per ottenere dettagli su un abbonamento eccezionale

L'esempio seguente fornisce dettagli su un abbonamento a Security Lake Exception. In questo esempio, l'utente dell'AWS account specificato riceve una notifica degli errori durante la SMS consegna. Il messaggio di eccezione rimane nell'account per il periodo di tempo specificato. Una sottoscrizione a eccezione notifica a un utente di Security Lake un errore tramite il protocollo preferito del richiedente.

```
aws securitylake get-data-lake-exception-subscription
```

Output:

```
{
  "exceptionTimeToLive": 30,
  "notificationEndpoint": "123456789012",
  "subscriptionProtocol": "sms"
}
```

Per ulteriori informazioni, consulta [Risoluzione dei problemi relativi allo stato del data lake](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [GetDataLakeExceptionSubscription AWS CLI Command Reference](#).

## get-data-lake-organization-configuration

Il seguente esempio di codice mostra come utilizzare `get-data-lake-organization-configuration`.

AWS CLI

Per ottenere dettagli sulla configurazione dei nuovi account dell'organizzazione

L'`get-data-lake-organization-configuration` seguente recupera i dettagli sui log di origine che i nuovi account dell'organizzazione invieranno dopo l'onboarding su Amazon Security Lake.

```
aws securitylake get-data-lake-organization-configuration
```

Output:

```
{
  "autoEnableNewAccount": [
    {
      "region": "us-east-1",
      "sources": [
        {
          "sourceName": "VPC_FLOW",
          "sourceVersion": "1.0"
        }
      ]
    }
  ]
}
```

```

    {
      "sourceName": "ROUTE53",
      "sourceVersion": "1.0"
    },
    {
      "sourceName": "SH_FINDINGS",
      "sourceVersion": "1.0"
    }
  ]
}

```

Per ulteriori informazioni, consulta [Managing multiple account with AWS Organizations](#) nella Amazon Security Lake User Guide.

- Per API i dettagli, consulta [GetDataLakeOrganizationConfiguration AWS CLI Command Reference](#).

## get-data-lake-sources

Il seguente esempio di codice mostra come utilizzare `get-data-lake-sources`.

### AWS CLI

Per ottenere lo stato della raccolta dei registri

L'esempio seguente ottiene un'istantanea della raccolta di log per l'account specificato nella AWS regione corrente. L'account ha Amazon Security Lake abilitato.

```
aws securitylake get-data-lake-sources \
  --accounts "123456789012"
```

Output:

```

{
  "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  
}
```

Per ulteriori informazioni, consulta [Raccolta di dati dai AWS servizi](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [GetDataLakeSources AWS CLI Command Reference](#).

## get-subscriber

Il seguente esempio di codice mostra come utilizzare `get-subscriber`.

### AWS CLI

Per recuperare le informazioni sull'abbonamento

L'`get-subscriber` seguente recupera le informazioni sull'abbonamento per l'abbonato Security Lake specificato.

```
aws securitylake get-subscriber \  
  --subscriber-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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-2-8ugsus4ztnsfjpbldwbgf4vge98av9/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"
}
}

```

Per ulteriori informazioni, consulta [Gestione degli abbonati](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [GetSubscriber AWS CLI](#) Command Reference.

## list-data-lake-exceptions

Il seguente esempio di codice mostra come utilizzare `list-data-lake-exceptions`.

### AWS CLI

Per elencare i problemi che riguardano il data lake

L'`list-data-lake-exceptions` seguente elenca i problemi che hanno interessato il data lake negli ultimi 14 giorni nelle AWS regioni specificate.

```
aws securitylake list-data-lake-exceptions \
  --regions "us-east-1" "eu-west-3"
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta la sezione [Risoluzione dei problemi di Amazon Security Lake](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [ListDataLakeExceptions AWS CLI Command Reference](#).

## list-data-lakes

Il seguente esempio di codice mostra come utilizzare `list-data-lakes`.

### AWS CLI

Per elencare l'oggetto di configurazione di Security Lake

L'`list-data-lakes` seguente elenca l'oggetto di configurazione Amazon Security Lake per la AWS regione specificata. È possibile utilizzare questo comando per determinare se Security Lake è abilitato in una o più regioni specificate.



```
aws securitylake list-data-lakes \  
--regions "us-east-1"
```

Output:

```
{  
  "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"  
      }  
    }  
  ]  
}
```

```
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Verifica dello stato della regione](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [ListDataLakes AWS CLI Command Reference](#).

## list-log-sources

Il seguente esempio di codice mostra come utilizzare `list-log-sources`.

### AWS CLI

Per recuperare le sorgenti di log di Amazon Security Lake

L'`list-log-sources` esempio seguente elenca le sorgenti di log di Amazon Security Lake in un account specificato.

```
aws securitylake list-log-sources \  
  --accounts "123456789012"
```

Output:

```
{  
  "account": "123456789012",  
  "region": "xy-region-1",  
  "sources": [  
    {  
      "awsLogSource": {  
        "sourceName": "VPC_FLOW",  
        "sourceVersion": "2.0"  
      }  
    },  
    {  
      "awsLogSource": {  
        "sourceName": "SH_FINDINGS",  
        "sourceVersion": "2.0"  
      }  
    }  
  ]  
}
```

```
]
}
```

Per ulteriori informazioni, consulta la sezione [Gestione del codice sorgente](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [ListLogSources AWS CLI Command Reference](#).

## list-subscribers

Il seguente esempio di codice mostra come utilizzare `list-subscribers`.

### AWS CLI

Per recuperare gli abbonati Amazon Security Lake

L'`list-subscribers` esempio seguente elenca tutti gli abbonati Amazon Security Lake in un account specifico.

```
aws securitylake list-subscribers
```

Output:

```
{
  "subscribers": [
    {
      "accessTypes": [
        "S3"
      ],
      "createdAt": "2024-06-04T15:02:28.921000+00:00",
      "roleArn": "arn:aws:iam::123456789012:role/AmazonSecurityLake-
E1WG1ZNPRT0D4",
      "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"
}
]
}

```

Per ulteriori informazioni, consulta [Gestione degli abbonati](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [ListSubscribers AWS CLI Command Reference](#).

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una risorsa esistente

L'`list-tags-for-resource` esempio seguente elenca i tag per l'abbonato Amazon Security Lake specificato. In questo esempio, la chiave del tag `Owner` non ha un valore di tag associato. È possibile utilizzare questa operazione per elencare i tag anche per altre risorse Security Lake esistenti.

```
aws securitylake list-tags-for-resource \  
  --resource-arn "arn:aws:securitylake:us-  
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab"
```

Output:

```
{  
  "tags": [  
    {  
      "key": "Environment",  
      "value": "Cloud"  
    },  
    {  
      "key": "CostCenter",  
      "value": "12345"  
    },  
    {  
      "key": "Owner",  
      "value": ""  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Taggare le risorse di Amazon Security Lake](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## register-data-lake-delegated-administrator

Il seguente esempio di codice mostra come utilizzare `register-data-lake-delegated-administrator`.

### AWS CLI

Per designare l'amministratore delegato

L'`register-data-lake-delegated-administrator` esempio seguente designa l' AWS account specificato come amministratore delegato di Amazon Security Lake.

```
aws securitylake register-data-lake-delegated-administrator \  
  --account-id 123456789012
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing multiple account with AWS Organizations](#) nella Amazon Security Lake User Guide.

- Per API i dettagli, consulta [RegisterDataLakeDelegatedAdministrator AWS CLI Command Reference](#).

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere tag a una risorsa esistente

L'`tag-resource` esempio seguente aggiunge tag a una risorsa di sottoscrizione esistente. Per creare una nuova risorsa e aggiungervi uno o più tag, non utilizzate questa operazione. Utilizzate invece l'operazione Create appropriata per il tipo di risorsa che desiderate creare.

```
aws securitylake tag-resource \  
  --resource-arn "arn:aws:securitylake:us-east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \  
  --tags key=Environment,value=Cloud
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare le risorse di Amazon Security Lake](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [TagResource AWS CLI](#) Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una risorsa esistente

L'`untag-resource` esempio seguente rimuove i tag specificati da una risorsa sottoscrittore esistente.

```
aws securitylake untag-resource \  
  --resource-arn "arn:aws:securitylake:us-  
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \  
  --tags Environment Owner
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Taggare le risorse di Amazon Security Lake](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [UntagResource AWS CLI](#) Command Reference.

## update-data-lake-exception-subscription

Il seguente esempio di codice mostra come utilizzare `update-data-lake-exception-subscription`.

### AWS CLI

Per aggiornare l'abbonamento alle notifiche per le eccezioni di Security Lake

L'`update-data-lake-exception-subscription` esempio seguente aggiorna la sottoscrizione alle notifiche che notifica agli utenti le eccezioni di Security Lake.

```
aws securitylake update-data-lake-exception-subscription \  
  --notification-endpoint "123456789012" \  
  --exception-time-to-live 30 \  
  \
```

```
--subscription-protocol "email"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Risoluzione dei problemi di Amazon Security Lake](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [UpdateDataLakeExceptionSubscription AWS CLI Command Reference](#).

## update-data-lake

Il seguente esempio di codice mostra come utilizzare `update-data-lake`.

### AWS CLI

Esempio 1: Per aggiornare le impostazioni del data lake

L'`update-data-lake`esempio seguente aggiorna le impostazioni del tuo data lake Amazon Security Lake. È possibile utilizzare questa operazione per specificare le impostazioni della regione di crittografia, archiviazione e rollup dei dati.

```
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"
```

Output:

```
{
  "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-
cehuifz15rwmhm6m62h7zhvtseogr9",
    "updateStatus": {
      "exception": {},
      "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
      "status": "INITIALIZED"
    }
  }
]
}

```

Per ulteriori informazioni, consulta la sezione [Guida introduttiva ad Amazon Security Lake](#) nella Guida per l'utente di Amazon Security Lake.

Esempio 2: configurare il data lake in una singola regione

L'create-data-lakeesempio seguente abilita Amazon Security Lake in una singola AWS regione e configura il tuo 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"

```

Output:

```

{
  "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"
      }
    }
  ]
}

```

Per ulteriori informazioni, consulta la sezione [Guida introduttiva ad Amazon Security Lake](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [UpdateDataLake AWS CLI Command Reference](#).

## update-subscriber-notification

Il seguente esempio di codice mostra come utilizzare `update-subscriber-notification`.

### AWS CLI

Per aggiornare una notifica di sottoscrizione

L'esempio seguente mostra come aggiornare il metodo di notifica per un abbonato.

```
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"}}'
```

Output:

```
{  
  "subscriberEndpoint": [  
    "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"  
  ]  
}
```

Per ulteriori informazioni, consulta [Gestione degli abbonati](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [UpdateSubscriberNotification AWS CLI Command Reference](#).

## update-subscriber

Il seguente esempio di codice mostra come utilizzare `update-subscriber`.

### AWS CLI

Per aggiornare un abbonato Amazon Security Lake.

L'esempio seguente aggiorna le fonti di accesso ai dati Security Lake per uno specifico abbonato a Security Lake.

```
aws securitylake update-subscriber \  

```

```
--subscriber-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{
  "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/
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/
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"
}
}

```

Per ulteriori informazioni, consulta [Gestione degli abbonati](#) nella Guida per l'utente di Amazon Security Lake.

- Per API i dettagli, consulta [UpdateSubscriber AWS CLI Command Reference](#).

## AWS Serverless Application Repository esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Serverless Application Repository.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **put-application-policy**

Il seguente esempio di codice mostra come utilizzare `put-application-policy`.

AWS CLI

Esempio 1: condividere un'applicazione pubblicamente

Quanto segue `put-application-policy` condivide un'applicazione pubblicamente, in modo che chiunque possa trovare e distribuire l'applicazione nel AWS Serverless Application Repository.

```
aws serverlessrepo put-application-policy \  
  --application-id arn:aws:serverlessrepo:us-east-1:123456789012:applications/my-  
test-application \  
  --statements Principals='*',Actions=Deploy
```

Output:

```
{  
  "Statements": [  
    {  
      "Actions": [  
        "Deploy"  
      ],  
      "Principals": [  
        ""  
      ],  
      "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

```

    }
  ]
}

```

## Esempio 2: condividere un'applicazione in privato

Quanto segue `put-application-policy` condivide un'applicazione privatamente, quindi solo AWS account specifici possono trovare e distribuire l'applicazione nel AWS Serverless Application Repository.

```

aws serverlessrepo put-application-policy \
  --application-id arn:aws:serverlessrepo:us-east-1:123456789012:applications/my-
test-application \
  --statements Principals=111111111111,222222222222,Actions=Deploy

```

## Output:

```

{
  "Statements": [
    {
      "Actions": [
        "Deploy"
      ],
      "Principals": [
        "111111111111",
        "222222222222"
      ],
      "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
    }
  ]
}

```

Per ulteriori informazioni, vedere [Condivisione di un'applicazione tramite la console](#) nella AWS Serverless Application Repository Developer Guide

- Per API i dettagli, vedere [PutApplicationPolicy](#) in AWS CLI Command Reference.

## Esempi di Service Catalog utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Service Catalog.



Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **accept-portfolio-share**

Il seguente esempio di codice mostra come utilizzare `accept-portfolio-share`.

AWS CLI

Accettare una quota di portafoglio

L'`accept-portfolio-share` esempio seguente accetta un'offerta, fatta da un altro utente, per condividere il portafoglio specificato.

```
aws servicecatalog accept-portfolio-share \
  --portfolio-id port-2s6wuabcdefghijk
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [AcceptPortfolioShare](#) in AWS CLI Command Reference.

### **associate-principal-with-portfolio**

Il seguente esempio di codice mostra come utilizzare `associate-principal-with-portfolio`.

AWS CLI

Associare un principale a un portafoglio

L'`associate-principal-with-portfolio` esempio seguente associa un utente al portafoglio specificato.

```
aws servicecatalog associate-principal-with-portfolio \  
  --portfolio-id port-2s6abcdefwdh4 \  
  --principal-arn arn:aws:iam::123456789012:user/usertest \  
  --principal-type IAM
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [AssociatePrincipalWithPortfolio](#) in AWS CLI Command Reference.

## associate-product-with-portfolio

Il seguente esempio di codice mostra come utilizzare `associate-product-with-portfolio`.

### AWS CLI

Associare un prodotto a un portafoglio

L'esempio seguente associa il prodotto dato al portafoglio specificato.

```
aws servicecatalog associate-product-with-portfolio \  
  --product-id prod-3p5abcdef3oyk \  
  --portfolio-id port-2s6abcdef5wdh4
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [AssociateProductWithPortfolio](#) in AWS CLI Command Reference.

## associate-tag-option-with-resource

Il seguente esempio di codice mostra come utilizzare `associate-tag-option-with-resource`.

### AWS CLI

Associare un a TagOption a una risorsa

L'esempio seguente associa la risorsa specificata TagOption alla risorsa specificata.

```
aws servicecatalog associate-tag-option-with-resource \  
  --resource-id port-2s6abcdq5wdh4 \  
  --tag-option-id port-2s6abcdq5wdh4
```

```
--tag-option-id tag-p3abc2pkpz5qc
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [AssociateTagOptionWithResource](#) in AWS CLI Command Reference.

## copy-product

Il seguente esempio di codice mostra come utilizzare `copy-product`.

### AWS CLI

Per copiare un prodotto

L'esempio seguente crea una copia del prodotto specificato, utilizzando un JSON file per passare i parametri.

```
aws servicecatalog copy-product --cli-input-json file://copy-product-input.json
```

Contenuto di `copy-product-input.json`.

```
{
  "SourceProductArn": "arn:aws:catalog:us-west-2:123456789012:product/prod-
tcabcd3syn2xy",
  "TargetProductName": "copy-of-myproduct",
  "CopyOptions": [
    "CopyTags"
  ]
}
```

Output:

```
{
  "CopyProductToken": "copyproduct-abc5defgjkdji"
}
```

- Per API i dettagli, vedere [CopyProduct](#) in AWS CLI Command Reference.

## create-portfolio-share

Il seguente esempio di codice mostra come utilizzare `create-portfolio-share`.

## AWS CLI

Per condividere un portafoglio con un account

L'`create-portfolio-share` seguente condivide il portafoglio specificato con il conto specificato.

```
aws servicecatalog create-portfolio-share \  
  --portfolio-id port-2s6abcdef5wdh4 \  
  --account-id 794123456789
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [CreatePortfolioShare](#) in AWS CLI Command Reference.

## `create-portfolio`

Il seguente esempio di codice mostra come utilizzare `create-portfolio`.

## AWS CLI

Per creare un portfolio

L'`create-portfolio` seguente crea un portfolio.

```
aws servicecatalog create-portfolio \  
  --provider-name my-provider \  
  --display-name my-portfolio
```

Output:

```
{  
  "PortfolioDetail": {  
    "ProviderName": "my-provider",  
    "DisplayName": "my-portfolio",  
    "CreatedTime": 1571337221.555,  
    "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/  
port-2s6xmplq5wdh4",  
    "Id": "port-2s6xmplq5wdh4"  
  }  
}
```

- Per API i dettagli, vedere [CreatePortfolio](#) in AWS CLI Command Reference.

## create-product

Il seguente esempio di codice mostra come utilizzare `create-product`.

### AWS CLI

Per creare un prodotto

L'esempio seguente crea un prodotto utilizzando un JSON file per passare i parametri.

```
aws servicecatalog create-product \  
  --cli-input-json file://create-product-input.json
```

Contenuto di `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"  
  }  
}
```

**Output:**

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [CreateProduct](#) in AWS CLI Command Reference.

**create-provisioning-artifact**

Il seguente esempio di codice mostra come utilizzare `create-provisioning-artifact`.

## AWS CLI

Per creare un artefatto di approvvigionamento

L'create-provisioning-artifactesempio seguente crea un artefatto di provisioning, utilizzando un file per passare i parametri. JSON

```
aws servicecatalog create-provisioning-artifact \  
  --cli-input-json file://create-provisioning-artifact-input.json
```

Contenuto di 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"  
  }  
}
```

Output:

```
{  
  "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  
  }  
}
```

- Per API i dettagli, vedere [CreateProvisioningArtifact](#) in AWS CLI Command Reference.

## create-tag-option

Il seguente esempio di codice mostra come utilizzare `create-tag-option`.

### AWS CLI

Per creare un TagOption

L'`create-tag-option`esempio seguente crea un TagOption.

```
aws servicecatalog create-tag-option
  --key 1234
  --value name
```

Output:

```
{
  "TagOptionDetail": {
    "Id": "tag-iabcdn4fzjjms",
    "Value": "name",
    "Active": true,
    "Key": "1234"
  }
}
```

- Per API i dettagli, vedere [CreateTagOption](#) in AWS CLI Command Reference.

## delete-portfolio-share

Il seguente esempio di codice mostra come utilizzare `delete-portfolio-share`.

### AWS CLI

Per interrompere la condivisione di un portafoglio con un account

L'`delete-portfolio-share`esempio seguente interrompe la condivisione del portafoglio con l'account specificato.

```
aws servicecatalog delete-portfolio-share \
  --portfolio-id port-2s6abcdq5wdh4 \
```



```
--account-id 123456789012
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeletePortfolioShare](#) in AWS CLI Command Reference.

## delete-portfolio

Il seguente esempio di codice mostra come utilizzare `delete-portfolio`.

### AWS CLI

Per eliminare un portfolio

L'`delete-portfolio` esempio seguente elimina il portfolio specificato.

```
aws servicecatalog delete-portfolio \  
  --id port-abcd1x4gox4do
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeletePortfolio](#) in AWS CLI Command Reference.

## delete-product

Il seguente esempio di codice mostra come utilizzare `delete-product`.

### AWS CLI

Per eliminare un prodotto

L'`delete-product` esempio seguente elimina il prodotto specificato.

```
aws servicecatalog delete-product \  
  --id prod-abcde6yhbxi
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteProduct](#) in AWS CLI Command Reference.

## delete-provisioning-artifact

Il seguente esempio di codice mostra come utilizzare `delete-provisioning-artifact`.

## AWS CLI

Per eliminare un elemento di approvvigionamento

L'`delete-provisioning-artifact`esempio seguente elimina l'artefatto di provisioning specificato.

```
aws servicecatalog delete-provisioning-artifact \  
  --product-id prod-abc2uebuplcpw \  
  --provisioning-artifact-id pa-pqabccddii7ouc
```

Questo comando non produce alcun output.

- Per i API dettagli, vedere [DeleteProvisioningArtifact](#)in Command Reference.AWS CLI

## `delete-tag-option`

Il seguente esempio di codice mostra come utilizzare`delete-tag-option`.

## AWS CLI

Per eliminare un TagOption

L'`delete-tag-option`esempio seguente elimina il valore specificato TagOption.

```
aws servicecatalog delete-tag-option \  
  --id tag-iabcdn4fzjjms
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteTagOption](#)in AWS CLI Command Reference.

## `describe-copy-product-status`

Il seguente esempio di codice mostra come utilizzare`describe-copy-product-status`.

## AWS CLI

Per descrivere lo stato dell'operazione di copia del prodotto

L'`describe-copy-product-status`esempio seguente visualizza lo stato corrente dell'operazione di copia asincrona specificata del prodotto.

```
aws servicecatalog describe-copy-product-status \  
  --copy-product-token copyproduct-znn5tf5abcd3w
```

Output:

```
{  
  "CopyProductStatus": "SUCCEEDED",  
  "TargetProductId": "prod-os6hog7abcdt2"  
}
```

- Per API i dettagli, vedere [DescribeCopyProductStatus](#) in AWS CLI Command Reference.

## describe-portfolio

Il seguente esempio di codice mostra come utilizzare `describe-portfolio`.

AWS CLI

Per descrivere un portfolio

L'`describe-portfolio` esempio seguente mostra i dettagli per il portafoglio specificato.

```
aws servicecatalog describe-portfolio \  
  --id port-2s6abcdq5wdh4
```

Output:

```
{  
  "TagOptions": [],  
  "PortfolioDetail": {  
    "ARN": "arn:aws:catalog:us-west-2:687558541234:portfolio/  
port-2s6abcdq5wdh4",  
    "Id": "port-2s6wuzyq5wdh4",  
    "CreatedTime": 1571337221.555,  
    "DisplayName": "my-portfolio",  
    "ProviderName": "my-provider"  
  },  
  "Tags": []  
}
```

- Per API i dettagli, vedere [DescribePortfolio](#) in AWS CLI Command Reference.

## describe-product-as-admin

Il seguente esempio di codice mostra come utilizzare `describe-product-as-admin`.

### AWS CLI

Descrivere un prodotto come amministratore

L'esempio seguente visualizza i dettagli del prodotto specificato utilizzando i privilegi di amministratore.

```
aws servicecatalog describe-product-as-admin \  
  --id prod-abcdcek6yhbx1
```

Output:

```
{  
  "TagOptions": [],  
  "ProductViewDetail": {  
    "ProductARN": "arn:aws:catalog:us-west-2:687558542028:product/prod-  
abcdcek6yhbx1",  
    "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-abcdcek6yhbx1",  
      "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"
    }
]
}

```

- Per API i dettagli, vedere [DescribeProductAsAdmin](#) in AWS CLI Command Reference.

## describe-provisioned-product

Il seguente esempio di codice mostra come utilizzare `describe-provisioned-product`.

### AWS CLI

Per descrivere un prodotto fornito

L'esempio seguente visualizza i dettagli del prodotto fornito specificato.

```

aws servicecatalog describe-provisioned-product \
  --id pp-dpom27bm4abcd

```

Output:

```

{
  "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-tfuawdjovzxge",
    "Type": "CFN_STACK",
    "Name": "mytestppname3"
  },
  "CloudWatchDashboards": []
}

```

- Per API i dettagli, vedere [DescribeProvisionedProduct](#) in AWS CLI Command Reference.

## describe-provisioning-artifact

Il seguente esempio di codice mostra come utilizzare `describe-provisioning-artifact`.

### AWS CLI

Per descrivere un elemento di approvvigionamento

L'`describe-provisioning-artifact` esempio seguente visualizza i dettagli per l'artefatto di provisioning specificato.

```

aws servicecatalog describe-provisioning-artifact \
  --provisioning-artifact-id pa-pcz347abcdcfm \
  --product-id prod-abcdfz3syn2rg

```

Output:

```

{
  "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"
}

```

- Per API i dettagli, vedere [DescribeProvisioningArtifact](#) in AWS CLI Command Reference.

## describe-tag-option

Il seguente esempio di codice mostra come utilizzare `describe-tag-option`.

### AWS CLI

Per descrivere un TagOption

L'`describe-tag-option`esempio seguente visualizza i dettagli relativi a quanto specificato TagOption.

```
aws servicecatalog describe-tag-option \  
  --id tag-p3tej2abcd5qc
```

Output:

```
{  
  "TagOptionDetail": {  
    "Active": true,  
    "Id": "tag-p3tej2abcd5qc",  
    "Value": "value-3",  
    "Key": "1234"  
  }  
}
```

- Per API i dettagli, vedere [DescribeTagOption](#) in AWS CLI Command Reference.

## disassociate-principal-from-portfolio

Il seguente esempio di codice mostra come utilizzare `disassociate-principal-from-portfolio`.

### AWS CLI

Dissociare un principale da un portafoglio

L'`disassociate-principal-from-portfolio`esempio seguente dissocia il capitale specificato dal portafoglio.

```
aws servicecatalog disassociate-principal-from-portfolio \  
  --portfolio-id port-2s6abcdq5wdh4 \  
  --principal-arn arn:aws:iam::123456789012:group/myendusers
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DisassociatePrincipalFromPortfolio](#) in AWS CLI Command Reference.

## disassociate-product-from-portfolio

Il seguente esempio di codice mostra come utilizzare `disassociate-product-from-portfolio`.

### AWS CLI

Dissociare un prodotto da un portafoglio

L'`disassociate-product-from-portfolio` esempio seguente dissocia il prodotto specificato dal portafoglio.

```
aws servicecatalog disassociate-product-from-portfolio \  
  --product-id prod-3p5abcdmu3oyk \  
  --portfolio-id port-2s6abcdq5wdh4
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DisassociateProductFromPortfolio](#) in AWS CLI Command Reference.

## disassociate-tag-option-from-resource

Il seguente esempio di codice mostra come utilizzare `disassociate-tag-option-from-resource`.

### AWS CLI

Dissociare un TagOption da una risorsa

L'`disassociate-tag-option-from-resource` esempio seguente dissocia la risorsa specificata TagOption dalla risorsa.

```
aws servicecatalog disassociate-tag-option-from-resource \  
  --resource-id port-2s6abcdq5wdh4 \  
  --tag-option-id port-2s6abcdq5wdh4
```



```
--tag-option-id tag-p3abc2pkpz5qc
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DisassociateTagOptionFromResource](#) in AWS CLI Command Reference.

## list-accepted-portfolio-shares

Il seguente esempio di codice mostra come utilizzare `list-accepted-portfolio-shares`.

### AWS CLI

Per elencare le azioni di portafoglio accettate

L'`list-accepted-portfolio-shares` esempio seguente elenca tutti i portafogli per i quali la condivisione è stata accettata da questo account, inclusi solo i portafogli predefiniti del Service Catalog.

```
aws servicecatalog list-accepted-portfolio-shares \
  --portfolio-share-type "AWS_SERVICECATALOG"
```

Output:

```
{
  "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"
    }
]
}
```

- Per API i dettagli, vedere [ListAcceptedPortfolioShares](#) in AWS CLI Command Reference.

## list-portfolio-access

Il seguente esempio di codice mostra come utilizzare `list-portfolio-access`.

### AWS CLI

Per elencare gli account con accesso a un portafoglio

L'`list-portfolio-access` seguente elenca gli AWS account che hanno accesso al portafoglio specificato.

```
aws servicecatalog list-portfolio-access \
  --portfolio-id port-2s6abcdq5wdh4
```

Output:

```
{
  "AccountIds": [
    "123456789012"
  ]
}
```

- Per API i dettagli, vedere [ListPortfolioAccess](#) in AWS CLI Command Reference.

## list-portfolios-for-product

Il seguente esempio di codice mostra come utilizzare `list-portfolios-for-product`.

### AWS CLI

Per elencare i portafogli associati a un prodotto

L'`list-portfolios-for-product` seguente elenca i portafogli associati al prodotto specificato.

```
aws servicecatalog list-portfolios-for-product \
  --product-id prod-abcdefz3syn2rg
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ListPortfoliosForProduct](#) in AWS CLI Command Reference.

## `list-portfolios`

Il seguente esempio di codice mostra come utilizzare `list-portfolios`.

### AWS CLI

Per elencare i portafogli

L'`list-portfolios` seguente elenca i portafogli Service Catalog nella regione corrente.

```
aws servicecatalog list-portfolios
```

Output:

```
{
  "PortfolioDetails": [
    {
      "CreatedTime": 1559665256.348,
      "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/
port-5pzcxmlst4ei",
      "DisplayName": "my-portfolio",
      "Id": "port-5pzcxmlst4ei",
      "ProviderName": "my-user"
    }
  ]
}
```

- Per API i dettagli, vedere [ListPortfolios](#) in AWS CLI Command Reference.

## list-principals-for-portfolio

Il seguente esempio di codice mostra come utilizzare `list-principals-for-portfolio`.

AWS CLI

Per elencare tutti i principi di un portafoglio

L'`list-principals-for-portfolio` esempio seguente elenca tutti i principali per il portafoglio specificato.

```
aws servicecatalog list-principals-for-portfolio \
  --portfolio-id port-2s6abcdq5wdh4
```

Output:

```
{
  "Principals": [
    {
      "PrincipalARN": "arn:aws:iam::123456789012:user/usertest",
      "PrincipalType": "IAM"
    }
  ]
}
```

```
]
}
```

- Per API i dettagli, vedere [ListPrincipalsForPortfolio](#) in AWS CLI Command Reference.

## list-provisioning-artifacts

Il seguente esempio di codice mostra come utilizzare `list-provisioning-artifacts`.

### AWS CLI

Per elencare tutti gli elementi di approvvigionamento per un prodotto

L'`list-provisioning-artifacts` esempio seguente elenca tutti gli artefatti di approvvigionamento per il prodotto specificato.

```
aws servicecatalog list-provisioning-artifacts \
  --product-id prod-nfi2abcdefgcpw
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ListProvisioningArtifacts](#) in AWS CLI Command Reference.

## list-resources-for-tag-option

Il seguente esempio di codice mostra come utilizzare `list-resources-for-tag-option`.

### AWS CLI

Per elencare le risorse associate a un TagOption

L'`list-resources-for-tag-option` esempio seguente elenca le risorse associate a quanto specificato TagOption.

```
aws servicecatalog list-resources-for-tag-option \
  --tag-option-id tag-p3tej2abcd5qc
```

Output:

```
{
  "ResourceDetails": [
    {
      "ARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-
abcdfz3syn2rg",
      "Name": "my product",
      "Description": "description",
      "CreatedTime": 1562097906.0,
      "Id": "prod-abcdfz3syn2rg"
    }
  ]
}
```

- Per API i dettagli, vedere [ListResourcesForTagOption](#) in AWS CLI Command Reference.

## list-tag-options

Il seguente esempio di codice mostra come utilizzare `list-tag-options`.

### AWS CLI

L'`list-tag-options` esempio seguente elenca tutti i valori per TagOptions.

```
aws servicecatalog list-tag-options
```

Output:

```
{
  "TagOptionDetails": [
    {
      "Value": "newvalue",
      "Active": true,
      "Id": "tag-iabcdn4fzjjms",
      "Key": "1234"
    },
    {
      "Value": "value1",
      "Active": true,
      "Id": "tag-e3abcdvmvvrzy",
      "Key": "key"
    }
  ]
}
```

- Per API i dettagli, vedere [ListTagOptions](#) in AWS CLI Command Reference.

## provision-product

Il seguente esempio di codice mostra come utilizzare `provision-product`.

AWS CLI

Fornire un prodotto

L'`provision-product` esempio seguente effettua il provisioning del prodotto specificato utilizzando l'artefatto di approvvigionamento specificato.

```
aws servicecatalog provision-product \
  --product-id prod-abcdfz3syn2rg \
  --provisioning-artifact-id pa-abc347pcscfm \
  --provisioned-product-name "mytestppname3"
```

Output:

```
{
  "RecordDetail": {
    "RecordId": "rec-tfuawdabcdege",
    "CreatedTime": 1577222793.362,
    "ProvisionedProductId": "pp-abcd27bm4mldq",
    "PathId": "lpv2-abcdg3jp6t5k6",
    "RecordErrors": [],
    "ProductId": "prod-abcdfz3syn2rg",
    "UpdatedTime": 1577222793.362,
    "RecordType": "PROVISION_PRODUCT",
    "ProvisionedProductName": "mytestppname3",
    "ProvisioningArtifactId": "pa-pcz347abcdcfm",
    "RecordTags": [],
    "Status": "CREATED",
    "ProvisionedProductType": "CFN_STACK"
  }
}
```

- Per API i dettagli, vedere [ProvisionProduct](#) in AWS CLI Command Reference.

## reject-portfolio-share

Il seguente esempio di codice mostra come utilizzare `reject-portfolio-share`.

### AWS CLI

Rifiutare una quota di portafoglio

L'esempio seguente rifiuta la quota di portafoglio per il portafoglio specificato.

```
aws servicecatalog reject-portfolio-share \
  --portfolio-id port-2s6wuabcdefghijk
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [RejectPortfolioShare](#) in AWS CLI Command Reference.

## scan-provisioned-products

Il seguente esempio di codice mostra come utilizzare `scan-provisioned-products`.



## AWS CLI

Per elencare tutti i prodotti forniti disponibili

L'`scan-provisioned-products` seguente elenca i prodotti forniti disponibili.

```
aws servicecatalog scan-provisioned-products
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ScanProvisionedProducts](#) in AWS CLI Command Reference.

## search-products-as-admin

Il seguente esempio di codice mostra come utilizzare `search-products-as-admin`.

## AWS CLI

Per cercare prodotti con privilegi di amministratore

L'`search-products-as-admin` seguente cerca prodotti con privilegi di amministratore, utilizzando un ID di portafoglio come filtro.

```
aws servicecatalog search-products-as-admin \  
  --portfolio-id port-5abcd3e5st4ei
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

- Per API i dettagli, consulta AWS CLI Command [SearchProductsAsAdmin](#) Reference.

## search-provisioned-products

Il seguente esempio di codice mostra come utilizzare `search-provisioned-products`.

### AWS CLI

Per cercare prodotti forniti

L'`search-provisioned-products` esempio seguente cerca i prodotti forniti che corrispondono all'ID del prodotto specificato, utilizzando un JSON file per passare i parametri.

```
aws servicecatalog search-provisioned-products \  
  --cli-input-json file://search-provisioned-products-input.json
```

## Contenuto di search-provisioned-products-input.json.

```
{
  "Filters": {
    "SearchQuery": [
      "prod-tcjevz3syn2rg"
    ]
  }
}
```

## Output:

```
{
  "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
}

```

- Per API i dettagli, vedere [SearchProvisionedProducts](#) in AWS CLI Command Reference.

## update-portfolio

Il seguente esempio di codice mostra come utilizzare `update-portfolio`.

### AWS CLI

Per aggiornare un portfolio

L'`update-portfolio` esempio seguente aggiorna il nome del portfolio specificato.

```

aws servicecatalog update-portfolio \
  --id port-5abcd3e5st4ei \
  --display-name "New portfolio name"

```

Output:

```

{
  "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": []
}

```

- Per API i dettagli, vedere [UpdatePortfolio](#) in AWS CLI Command Reference.

## update-product

Il seguente esempio di codice mostra come utilizzare `update-product`.

### AWS CLI

Per aggiornare un prodotto

L'`update-product` esempio seguente aggiorna il nome e il proprietario del prodotto specificato.

```

aws servicecatalog update-product \
  --id prod-os6abc7drqlt2 \
  --name "New product name" \
  --owner "Updated product owner"

```

Output:

```

{
  "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-6abcdgrfhhvidy",
        "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
}
}

```

- Per API i dettagli, vedere [UpdateProduct](#) in AWS CLI Command Reference.

## update-provisioning-artifact

Il seguente esempio di codice mostra come utilizzare `update-provisioning-artifact`.

### AWS CLI

Per aggiornare un elemento di provisioning

L'update-provisioning-artifactesempio seguente aggiorna il nome e la descrizione dell'elemento di provisioning specificato, utilizzando un file per passare i parametri. JSON

```
aws servicecatalog update-provisioning-artifact \
  --cli-input-json file://update-provisioning-artifact-input.json
```

Contenuto di `update-provisioning-artifact-input.json`.

```
{
  "ProductId": "prod-abcdefz3syn2rg",
  "ProvisioningArtifactId": "pa-pcz347abcdcfm",
  "Name": "updated name",
  "Description": "updated description"
}
```

Output:

```
{
```

```
"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
}
}
```

- Per API i dettagli, vedere [UpdateProvisioningArtifact](#) in AWS CLI Command Reference.

## update-tag-option

Il seguente esempio di codice mostra come utilizzare `update-tag-option`.

### AWS CLI

Per aggiornare un TagOption

L'update-tag-optionesempio seguente aggiorna il valore di aTagOption, utilizzando il JSON file specificato.

```
aws servicecatalog update-tag-option --cli-input-json file://update-tag-option-
input.json
```

Contenuto di `update-tag-option-input.json`.

```
{
  "Id": "tag-iabcdn4fzjjms",
  "Value": "newvalue",
  "Active": true
}
```

Output:

```
{
```

```
"TagOptionDetail": {
  "Value": "newvalue",
  "Key": "1234",
  "Active": true,
  "Id": "tag-iabcdn4fzjjms"
}
```

- Per API i dettagli, vedere [UpdateTagOption](#) in AWS CLI Command Reference.

## Esempi di Service Quotas utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Service Quotas.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **get-aws-default-service-quota**

Il seguente esempio di codice mostra come utilizzare `get-aws-default-service-quota`.

#### AWS CLI

Per descrivere una quota di servizio predefinita

L'`get-aws-default-service-quota` esempio seguente visualizza i dettagli per la quota specificata.

```
aws service-quotas get-aws-default-service-quota \
  --service-code ec2 \
```



```
--quota-code L-1216C47A
```

Output:

```
{
  "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"
    }
  }
}
```

- Per API i dettagli, vedere [GetAwsDefaultServiceQuota](#) in AWS CLI Command Reference.

## get-requested-service-quota-change

Il seguente esempio di codice mostra come utilizzare `get-requested-service-quota-change`.

AWS CLI

Per descrivere una richiesta di aumento della quota di servizio

L'esempio seguente descrive la richiesta di aumento della quota specificata.

```
aws service-quotas get-requested-service-quota-change \
  --request-id d187537d15254312a9609aa51bbf7624u7W49tP0
```

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, vedere [GetRequestedServiceQuotaChange](#) in AWS CLI Command Reference.

## get-service-quota

Il seguente esempio di codice mostra come utilizzare `get-service-quota`.

### AWS CLI

Per descrivere una quota di servizio

L'`get-service-quota` esempio seguente visualizza i dettagli sulla quota specificata.

```
aws service-quotas get-service-quota \
  --service-code ec2 \
  --quota-code L-1216C47A
```

**Output:**

```
{
  "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"
    }
  }
}
```

- Per API i dettagli, vedere [GetServiceQuota](#) in AWS CLI Command Reference.

**list-aws-default-service-quotas**

Il seguente esempio di codice mostra come utilizzare `list-aws-default-service-quotas`.

**AWS CLI**

Per elencare le quote predefinite per un servizio

L'`list-aws-default-service-quotas` esempio seguente elenca i valori predefiniti per le quote per il servizio specificato.

```
aws service-quotas list-aws-default-service-quotas \
```

```
--service-code xray
```

Output:

```
{
  "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
    }
  ]
}
```

- Per API i dettagli, vedere [ListAwsDefaultServiceQuotas](#) in AWS CLI Command Reference.

## list-requested-service-quota-change-history-by-quota

Il seguente esempio di codice mostra come utilizzare `list-requested-service-quota-change-history-by-quota`.

### AWS CLI

Per elencare le richieste di aumento della quota

L'`list-requested-service-quota-change-history-by-quota` esempio seguente elenca le richieste di aumento della quota per la quota specificata.

```
aws service-quotas list-requested-service-quota-change-history-by-quota \
  --service-code ec2 \
  --quota-code L-20F13EBD
```

### Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ListRequestedServiceQuotaChangeHistoryByQuota](#) in AWS CLI Command Reference.

## list-requested-service-quota-change-history

Il seguente esempio di codice mostra come utilizzare `list-requested-service-quota-change-history`.

### AWS CLI

Per elencare le richieste di aumento della quota

L'`list-requested-service-quota-change-history`esempio seguente elenca le richieste di aumento della quota per il servizio specificato.

```
aws service-quotas list-requested-service-quota-change-history \
  --service-code ec2
```

### Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ListRequestedServiceQuotaChangeHistory](#) in AWS CLI Command Reference.

## list-service-quotas

Il seguente esempio di codice mostra come utilizzare `list-service-quotas`.

### AWS CLI

Per elencare le quote per un servizio

L'esempio seguente visualizza i dettagli sulle quote per AWS CloudFormation

```
aws service-quotas list-service-quotas \
  --service-code cloudformation
```

Output:

```
{
  "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
    }
  ]
}
```

```
}
```

- Per API i dettagli, vedere [ListServiceQuotas](#) in AWS CLI Command Reference.

## list-services

Il seguente esempio di codice mostra come utilizzare `list-services`.

### AWS CLI

Per elencare i servizi disponibili

Il comando seguente elenca i servizi disponibili in Service Quotas.

```
aws service-quotas list-services
```

Output:

```
{
  "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"
    }
  ]
}
```



Puoi aggiungere il `--query` parametro per filtrare la visualizzazione in base alle informazioni che ti interessano. L'esempio seguente visualizza solo i codici di servizio.

```
aws service-quotas list-services \  
  --query Services[*].ServiceCode
```

Output:

```
[  
  "AWSCloudMap",  
  "access-analyzer",  
  "acm",  
  "acm-pca",  
  "amplify",  
  "apigateway",  
  "application-autoscaling",  
  ...truncated...  
  "xray"  
]
```

- Per API i dettagli, vedere [ListServices](#) in AWS CLI Command Reference.

## request-service-quota-increase

Il seguente esempio di codice mostra come utilizzare `request-service-quota-increase`.

AWS CLI

Per richiedere un aumento della quota di servizio

L'esempio seguente richiede un aumento della quota di servizio specificata.

```
aws service-quotas request-service-quota-increase \  
  --service-code ec2 \  
  --quota-code L-20F13EBD \  
  --desired-value 2
```

Output:

```
{
```

```
"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"
}
```

- Per API i dettagli, vedere [RequestServiceQuotaIncrease](#) in AWS CLI Command Reference.

## SESEsempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con AmazonSES.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, dove puoi trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **delete-identity**

Il seguente esempio di codice mostra come utilizzare `delete-identity`.

## AWS CLI

### Eliminazione di un'identità

L'esempio seguente utilizza il `delete-identity` comando per eliminare un'identità dall'elenco di identità verificate con AmazonSES:

```
aws ses delete-identity --identity user@example.com
```

Per ulteriori informazioni sulle identità verificate, consulta [Verifying Email Addresses and Domains in Amazon](#) nella Amazon SES Simple Email Service Developer Guide.

- Per API i dettagli, consulta [Command `DeleteIdentity` Reference AWS CLI](#) .

## get-identity-dkim-attributes

Il seguente esempio di codice mostra come utilizzare `get-identity-dkim-attributes`.

### AWS CLI

Per ottenere gli DKIM attributi di Amazon SES Easy per un elenco di identità

L'esempio seguente utilizza il `get-identity-dkim-attributes` comando per recuperare gli DKIM attributi di Amazon SES Easy per un elenco di identità:

```
aws ses get-identity-dkim-attributes --identities "example.com" "user@example.com"
```

Output:

```
{
  "DkimAttributes": {
    "example.com": {
      "DkimTokens": [
        "EXAMPLEejcs5xoyqytjsotsijas7236gr",
        "EXAMPLEejr76cvoc6mysspnioorxsn6ep",
        "EXAMPLEkbnkqkhlm2lyz77ppkulerm4k"
      ],
      "DkimEnabled": true,
      "DkimVerificationStatus": "Success"
    },
    "user@example.com": {
      "DkimEnabled": false,
      "DkimVerificationStatus": "NotStarted"
    }
  }
}
```

```
    }  
  }  
}
```

Se chiami questo comando con un'identità che non hai mai inviato per la verifica, tale identità non verrà visualizzata nell'output.

Per ulteriori informazioni su EasyDKIM, consulta Easy DKIM in Amazon SES nella Amazon Simple Email Service Developer Guide.

- Per API i dettagli, consulta [GetIdentityDkimAttributes AWS CLI Command Reference](#).

## get-identity-notification-attributes

Il seguente esempio di codice mostra come utilizzare `get-identity-notification-attributes`.

### AWS CLI

Per ottenere gli attributi di SES notifica di Amazon per un elenco di identità

L'esempio seguente utilizza il `get-identity-notification-attributes` comando per recuperare gli attributi di SES notifica di Amazon per un elenco di identità:

```
aws ses get-identity-notification-attributes --  
identities "user1@example.com" "user2@example.com"
```

Output:

```
{  
  "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  
    }  
  }  
}
```

Questo comando restituisce lo stato dell'inoltro del feedback via e-mail e, se applicabile, gli Amazon Resource Names (ARNs) degli SNS argomenti Amazon a cui vengono inviate le notifiche di rimbalzo, reclamo e consegna.

Se chiami questo comando con un'identità che non hai mai inviato per la verifica, tale identità non verrà visualizzata nell'output.

Per ulteriori informazioni sulle notifiche, consulta [Using Notifications With Amazon SES](#) nella [Amazon Simple Email Service Developer Guide](#).

- Per API i dettagli, consulta [GetIdentityNotificationAttributes AWS CLI Command Reference](#).

## get-identity-verification-attributes

Il seguente esempio di codice mostra come utilizzare `get-identity-verification-attributes`.

### AWS CLI

Per ottenere lo stato SES di verifica di Amazon per un elenco di identità

L'esempio seguente utilizza il `get-identity-verification-attributes` comando per recuperare lo stato di SES verifica di Amazon per un elenco di identità:

```
aws ses get-identity-verification-attributes --
identities "user1@example.com" "user2@example.com"
```

Output:

```
{
  "VerificationAttributes": {
    "user1@example.com": {
      "VerificationStatus": "Success"
    },
    "user2@example.com": {
      "VerificationStatus": "Pending"
    }
  }
}
```

Se chiami questo comando con un'identità che non hai mai inviato per la verifica, tale identità non verrà visualizzata nell'output.

Per ulteriori informazioni sulle identità verificate, consulta [Verifying Email Addresses and Domains](#) in Amazon nella Amazon SES Simple Email Service Developer Guide.

- Per API i dettagli, consulta [Command `GetIdentityVerificationAttributes` Reference AWS CLI](#).

## get-send-quota

Il seguente esempio di codice mostra come utilizzare `get-send-quota`.

### AWS CLI

Per stabilire i limiti di SES invio di Amazon

L'esempio seguente utilizza il `get-send-quota` comando per restituire i limiti di SES invio di Amazon:

```
aws ses get-send-quota
```

Output:

```
{
  "Max24HourSend": 200.0,
  "SentLast24Hours": 1.0,
  "MaxSendRate": 1.0
}
```

`Max24 HourSend` è la tua quota di invio, ovvero il numero massimo di e-mail che puoi inviare in un periodo di 24 ore. La quota di invio riflette un periodo di tempo continuo. Ogni volta che provi a inviare un'e-mail, Amazon SES controlla quante e-mail hai inviato nelle 24 ore precedenti. Se il numero totale di e-mail che hai inviato è inferiore alla quota, la tua richiesta sarà accettata e l'e-mail inviata.

`SentLast24 ore` è il numero di e-mail che hai inviato nelle 24 ore precedenti.

`MaxSendRate` è il numero massimo di e-mail che puoi inviare al secondo.

Tieni presente che i limiti di invio si basano sui destinatari e non sui messaggi. Ad esempio, un'e-mail con 10 destinatari viene conteggiata come 10 e-mail ai fini della quota sugli invii.

Per ulteriori informazioni, consulta [Managing Your Amazon SES Sending Limits](#) nella Amazon Simple Email Service Developer Guide.

- Per API i dettagli, consulta [GetSendQuota AWS CLI Command Reference](#).

## get-send-statistics

Il seguente esempio di codice mostra come utilizzare `get-send-statistics`.

### AWS CLI

Per ottenere le statistiche di SES invio su Amazon

L'esempio seguente utilizza il `get-send-statistics` comando per restituire le statistiche di SES invio di Amazon

```
aws ses get-send-statistics
```

Output:

```
{
  "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
    }
  ]
}
```

Il risultato è un elenco di punti dati, che rappresentano le ultime due settimane di attività di invio. Ogni punto dati nell'elenco contiene statistiche per un intervallo di 15 minuti.

In questo esempio, ci sono solo due punti dati perché le uniche e-mail inviate dall'utente nelle ultime due settimane rientrano in due intervalli di 15 minuti.

Per ulteriori informazioni, consulta [Monitoring Your Amazon SES Usage Statistics](#) nella [Amazon Simple Email Service Developer Guide](#).

- Per API i dettagli, consulta [GetSendStatistics AWS CLI Command Reference](#).

## list-identities

Il seguente esempio di codice mostra come utilizzare `list-identities`.

### AWS CLI

Per elencare tutte le identità (indirizzi e-mail e domini) per un account specifico AWS

L'esempio seguente utilizza il `list-identities` comando per elencare tutte le identità che sono state inviate per la verifica con AmazonSES:

```
aws ses list-identities
```

Output:

```
{
  "Identities": [
    "user@example.com",
    "example.com"
  ]
}
```

L'elenco restituito contiene tutte le identità indipendentemente dallo stato della verifica (verified, pending verification, failure, ecc.).

In questo esempio, gli indirizzi e-mail e i domini vengono restituiti perché non è stato specificato il parametro `identity-type`.

Per ulteriori informazioni sulla verifica, consulta [Verifying Email Addresses and Domains in Amazon SES](#) nella [Amazon Simple Email Service Developer Guide](#).

- Per API i dettagli, consulta [AWS CLI Command ListIdentities Reference](#).

## send-email

Il seguente esempio di codice mostra come utilizzare `send-email`.

### AWS CLI

Per inviare un'e-mail formattata tramite Amazon SES



Nell'esempio seguente viene utilizzato il comando `send-email` per inviare un messaggio e-mail formattato:

```
aws ses send-email --from sender@example.com --destination file://destination.json
--message file://message.json
```

Output:

```
{
  "MessageId": "EXAMPLEf3a5efcd1-51adec81-d2a4-4e3f-9fe2-5d85c1b23783-000000"
}
```

La destinazione e il messaggio sono strutture di JSON dati salvate in `file.json` nella directory corrente. Tali file sono i seguenti:

`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"
    }
  }
}
```

```
}  
}
```

Sostituisci gli indirizzi e-mail del mittente e del destinatario con quelli che desideri utilizzare. Tieni presente che l'indirizzo e-mail del mittente deve essere verificato con AmazonSES. Fino a quando non ti viene concesso l'accesso alla produzione ad AmazonSES, devi anche verificare l'indirizzo e-mail di ciascun destinatario, a meno che il destinatario non sia il simulatore di caselle di SES posta Amazon. Per ulteriori informazioni sulla verifica, consulta [Verifying Email Addresses and Domains in Amazon SES](#) nella [Amazon Simple Email Service Developer Guide](#).

L'ID del messaggio nell'output indica che la chiamata a `send-email` è stata completata correttamente.

Se non ricevi l'e-mail, controlla la casella della posta indesiderata.

Per ulteriori informazioni sull'invio di e-mail formattate, consulta [Invio di e-mail formattate utilizzando Amazon SES API](#) nella [Amazon Simple Email Service Developer Guide](#).

- Per API i dettagli, consulta AWS CLI Command [SendEmail](#)Reference.

## **send-raw-email**

Il seguente esempio di codice mostra come utilizzare `send-raw-email`.

### AWS CLI

Per inviare un'e-mail non elaborata tramite Amazon SES

L'esempio seguente utilizza il `send-raw-email` comando per inviare un'e-mail con un TXT allegato:

```
aws ses send-raw-email --raw-message file://message.json
```

Output:

```
{  
  "MessageId": "EXAMPLEf3f73d99b-c63fb06f-d263-41f8-a0fb-d0dc67d56c07-000000"  
}
```

Il messaggio non elaborato è una struttura di JSON dati salvata in un file denominato `message.json` nella directory corrente. Contiene i seguenti dati:

```
{
  "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--"
}
```

Come puoi vedere, «Data» è una lunga stringa che contiene l'intero contenuto non elaborato dell'e-mail in MIME formato, incluso un allegato chiamato attachment.txt.

Sostituisci sender@example.com e recipient@example.com con gli indirizzi che desideri utilizzare. Tieni presente che l'indirizzo e-mail del mittente deve essere verificato con AmazonSES. Fino a quando non ti viene concesso l'accesso alla produzione ad AmazonSES, devi anche verificare l'indirizzo e-mail del destinatario, a meno che il destinatario non sia il simulatore di caselle di SES posta Amazon. Per ulteriori informazioni sulla verifica, consulta [Verifying Email Addresses and Domains in Amazon SES](#) nella Amazon Simple Email Service Developer Guide.

L'ID del messaggio nell'output indica che la chiamata a send-raw-email è andata a buon fine.

Se non ricevi l'e-mail, controlla la casella della posta indesiderata.

Per ulteriori informazioni sull'invio di e-mail non elaborate, consulta [Sending Raw Email Using the Amazon SES API](#) nella Amazon Simple Email Service Developer Guide.

- Per API i dettagli, consulta [SendRawEmail AWS CLI Command Reference](#).

## set-identity-dkim-enabled

Il seguente esempio di codice mostra come utilizzare set-identity-dkim-enabled.

### AWS CLI

Per abilitare o disabilitare l'identità SES verificata di Easy DKIM for an Amazon

L'esempio seguente utilizza il set-identity-dkim-enabled comando di disabilitazione DKIM per un indirizzo email verificato:

```
aws ses set-identity-dkim-enabled --identity user@example.com --no-dkim-enabled
```

Per ulteriori informazioni su EasyDKIM, consulta Easy DKIM in Amazon SES nella Amazon Simple Email Service Developer Guide.

- Per API i dettagli, consulta [SetIdentityDkimEnabled AWS CLI Command Reference](#).

## **set-identity-feedback-forwarding-enabled**

Il seguente esempio di codice mostra come utilizzare `set-identity-feedback-forwarding-enabled`.

### AWS CLI

Per abilitare o disabilitare l'inoltro di feedback via email di rimbalzo e reclamo per un'identità verificata da Amazon SES

L'esempio seguente utilizza il `set-identity-feedback-forwarding-enabled` comando per consentire a un indirizzo e-mail verificato di ricevere notifiche di respingimento e reclamo via e-mail:

```
aws ses set-identity-feedback-forwarding-enabled --identity user@example.com --  
forwarding-enabled
```

Devi ricevere notifiche di rimbalzo e reclamo tramite Amazon SNS o l'inoltro di feedback via e-mail, quindi puoi disabilitare l'inoltro di feedback via e-mail solo se selezioni un SNS argomento Amazon sia per le notifiche di rimbalzo che per quelle di reclamo.

Per ulteriori informazioni sulle notifiche, consulta Using Notifications With Amazon SES nella Amazon Simple Email Service Developer Guide.

- Per API i dettagli, consulta [SetIdentityFeedbackForwardingEnabled AWS CLI Command Reference](#).

## **set-identity-notification-topic**

Il seguente esempio di codice mostra come utilizzare `set-identity-notification-topic`.

### AWS CLI

Per impostare l'SNS argomento Amazon su cui Amazon SES pubblicherà notifiche di rimbalzo, reclamo e/o consegna per un'identità verificata

L'esempio seguente utilizza il `set-identity-notification-topic` comando per specificare l'SNS argomento Amazon a cui un indirizzo e-mail verificato riceverà le notifiche di rimbalzo:

```
aws ses set-identity-notification-topic --identity user@example.com --notification-type Bounce --sns-topic arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic
```

Per ulteriori informazioni sulle notifiche, consulta Using Notifications With Amazon SES nella Amazon Simple Email Service Developer Guide.

- Per API i dettagli, consulta [SetIdentityNotificationTopic AWS CLI Command Reference](#).

## verify-domain-dkim

Il seguente esempio di codice mostra come utilizzare `verify-domain-dkim`.

### AWS CLI

Per generare i DKIM token di un dominio verificato per la DKIM firma con Amazon SES

L'esempio seguente utilizza il `verify-domain-dkim` comando per generare DKIM token per un dominio che è stato verificato con AmazonSES:

```
aws ses verify-domain-dkim --domain example.com
```

Output:

```
{
  "DkimTokens": [
    "EXAMPLEEq76owjnks31nluwg65scbemvw",
    "EXAMPLEi3dnsj67hstzaj673klariwx2",
    "EXAMPLEwfbtcukvimehexktmdtaz6naj"
  ]
}
```

Per eseguire la configurazioneDKIM, devi utilizzare i DKIM token restituiti per aggiornare DNS le impostazioni del tuo dominio con CNAME record che puntano a chiavi DKIM pubbliche ospitate da AmazonSES. Per ulteriori informazioni, consulta Easy DKIM in Amazon SES nella Amazon Simple Email Service Developer Guide.

- Per API i dettagli, consulta [VerifyDomainDkim AWS CLI Command Reference](#).

## verify-domain-identity

Il seguente esempio di codice mostra come utilizzare `verify-domain-identity`.

### AWS CLI

Per verificare un dominio con Amazon SES

Nell'esempio seguente viene utilizzato il comando `verify-domain-identity` per verificare un dominio:

```
aws ses verify-domain-identity --domain example.com
```

Output:

```
{
  "VerificationToken": "eoEmxw+YaYhb3h3iVJHuXMJXqeu1q1/wmvjuEXAMPLE"
}
```

Per completare la verifica del dominio, devi aggiungere un TXT record con il token di verifica restituito alle DNS impostazioni del dominio. Per ulteriori informazioni, consulta [Verifying Domains in Amazon SES](#) nella [Amazon Simple Email Service Developer Guide](#).

- Per API i dettagli, consulta AWS CLI Command [VerifyDomainIdentity](#) Reference.

## verify-email-identity

Il seguente esempio di codice mostra come utilizzare `verify-email-identity`.

### AWS CLI

Per verificare un indirizzo e-mail con Amazon SES

Nell'esempio seguente viene utilizzato il comando `verify-email-identity` per verificare un indirizzo e-mail:

```
aws ses verify-email-identity --email-address user@example.com
```

Prima di poter inviare un'e-mail tramite AmazonSES, devi verificare l'indirizzo o il dominio da cui stai inviando l'e-mail per dimostrare che la possiedi. Se non disponi ancora dell'accesso alla produzione, devi anche verificare tutti gli indirizzi e-mail a cui invii e-mail, ad eccezione degli indirizzi e-mail forniti dal simulatore di SES casella di posta Amazon.

Dopo `verify-email-identity` la chiamata, l'indirizzo e-mail riceverà un'e-mail di verifica. L'utente deve fare clic sul link nell'e-mail per completare il processo di verifica.

Per ulteriori informazioni, consulta la sezione *Verifica degli indirizzi e-mail* in Amazon SES nella *Amazon Simple Email Service Developer Guide*.

- Per API i dettagli, consulta [VerifyEmailIdentity AWS CLI Command Reference](#).

## Esempi di utilizzo di Shield AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Shield.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **associate-drt-log-bucket**

Il seguente esempio di codice mostra come utilizzare `associate-drt-log-bucket`.

AWS CLI

Per autorizzare l'accesso DRT a un bucket Amazon S3

L'esempio seguente crea un'associazione tra il bucket S3 DRT e il bucket S3 specificato. Ciò consente di accedere DRT al bucket per conto dell'account. :

```
aws shield associate-drt-log-bucket \  
  --log-bucket flow-logs-for-website-lb
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Authorize the DDoS Response Team](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [AssociateDrtLogBucket AWS CLI](#) Command Reference.

## **associate-drt-role**

Il seguente esempio di codice mostra come utilizzare `associate-drt-role`.

### AWS CLI

Per autorizzarli DRT a mitigare potenziali attacchi per conto dell'utente

L'esempio seguente crea un'associazione tra il ruolo DRT e il ruolo specificato. DRT possono utilizzare il ruolo per accedere e gestire l'account.

```
aws shield associate-drt-role \  
  --role-arn arn:aws:iam::123456789012:role/service-role/DrtRole
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Authorize the DDoS Response Team](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [AssociateDrtRole AWS CLI](#) Command Reference.

## **create-protection**

Il seguente esempio di codice mostra come utilizzare `create-protection`.

### AWS CLI

Per abilitare la protezione AWS Shield Advanced per una singola AWS risorsa

L'esempio seguente abilita la protezione Shield Advanced per la AWS CloudFront distribuzione specificata.

```
aws shield create-protection \  
  --name "Protection for CloudFront distribution" \  
  --resource-arn arn:aws:cloudfront::123456789012:distribution/E198WC25FX0WY8
```

Output:



```
{
  "ProtectionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Per ulteriori informazioni, consulta [Specificare le risorse da proteggere](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [CreateProtection AWS CLI](#) Command Reference.

## create-subscription

Il seguente esempio di codice mostra come utilizzare `create-subscription`.

### AWS CLI

Per abilitare la protezione AWS Shield Advanced per un account

L'`create-subscription` esempio seguente abilita la protezione Shield Advanced per l'account.

```
aws shield create-subscription
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Getting Started with AWS Shield Advanced](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [CreateSubscription AWS CLI](#) Command Reference.

## delete-protection

Il seguente esempio di codice mostra come utilizzare `delete-protection`.

### AWS CLI

Per rimuovere la protezione AWS Shield Advanced da una AWS risorsa

L'`delete-protection` esempio seguente rimuove la protezione AWS Shield Advanced specificata.

```
aws shield delete-protection \
  --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Rimuovere AWS Shield Advanced da una AWS risorsa](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [DeleteProtection AWS CLI](#) Command Reference.

## describe-attack

Il seguente esempio di codice mostra come utilizzare `describe-attack`.

### AWS CLI

Per recuperare una descrizione dettagliata di un attacco

L'`describe-attack` esempio seguente visualizza i dettagli sull'DDoS attacco con l'ID di attacco specificato. È possibile ottenere un attacco IDs eseguendo il `list-attacks` comando.

```
aws shield describe-attack --attack-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Output:

```
{
  "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": []
}
}
```

Per ulteriori informazioni, consulta Review [DDoSIncidents](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [DescribeAttack](#) in AWS CLI Command Reference.

## describe-drt-access

Il seguente esempio di codice mostra come utilizzare `describe-drt-access`.

### AWS CLI

Per recuperare una descrizione delle autorizzazioni, DRT deve mitigare gli attacchi per tuo conto

L'esempio seguente recupera le autorizzazioni relative al ruolo e al bucket S3 di cui DRT dispone, che gli consentono di rispondere a potenziali attacchi da parte dell'utente.

```
aws shield describe-drt-access
```

Output:

```
{
  "RoleArn": "arn:aws:iam::123456789012:role/service-role/DrtRole",
  "LogBucketList": [
    "flow-logs-for-website-lb"
  ]
}
```

Per ulteriori informazioni, consulta [Authorize the DDoS Response Team](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [DescribeDrtAccess AWS CLI](#) Command Reference.

## describe-emergency-contact-settings

Il seguente esempio di codice mostra come utilizzare `describe-emergency-contact-settings`.

### AWS CLI

Per recuperare gli indirizzi e-mail di emergenza archiviati presso il DRT

L'esempio seguente recupera gli indirizzi di posta elettronica registrati presso DRT l'account. Questi sono gli indirizzi a cui DRT rivolgersi per rispondere a un attacco sospetto.

```
aws shield describe-emergency-contact-settings
```

Output:

```
{
  "EmergencyContactList": [
    {
      "EmailAddress": "ops@example.com"
    },
    {
      "EmailAddress": "ddos-notifications@example.com"
    }
  ]
}
```

Per ulteriori informazioni, consulta [How AWS Shield Works](https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html) nella Shield Advanced Developer Guide.



- Per i dettagli, consulta Command Reference. API [DescribeEmergencyContactSettings](#) AWS CLI

## describe-protection

Il seguente esempio di codice mostra come utilizzare `describe-protection`.

### AWS CLI

Per recuperare i dettagli per una AWS protezione Shield Advanced

L'esempio seguente visualizza i dettagli sulla protezione Shield Advanced con l'ID specificato. È possibile ottenere protezione IDs eseguendo il `list-protections` comando.

```
aws shield describe-protection \  
  --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "Protection": {  
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",  
    "Name": "1.2.3.4",  
    "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:eip-allocation/  
eipalloc-0ac1537af40742a6d"  
  }  
}
```

Per ulteriori informazioni, consulta [Specificare le risorse da proteggere](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [DescribeProtection AWS CLI](#) Command Reference.

## describe-subscription

Il seguente esempio di codice mostra come utilizzare `describe-subscription`.

### AWS CLI

Per recuperare i dettagli della protezione AWS Shield Advanced per l'account

L'output di `aws shield describe-subscription` seguente mostra i dettagli sulla protezione Shield Advanced fornita per l'account. :

```
aws shield describe-subscription
```

Output:

```
{
  "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
      }
    ]
  }
}
```

Per ulteriori informazioni, consulta [How AWS Shield Works](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [DescribeSubscription AWS CLI Command Reference](#).

## disassociate-drt-log-bucket

Il seguente esempio di codice mostra come utilizzare `disassociate-drt-log-bucket`.

### AWS CLI

Per rimuovere l'autorizzazione all'accesso DRT a un bucket Amazon S3 per tuo conto

L'`disassociate-drt-log-bucket` esempio seguente rimuove l'associazione tra il bucket S3 DRT e il bucket S3 specificato. Una volta completato questo comando, non DRT possono più accedere al bucket per conto dell'account.

```
aws shield disassociate-drt-log-bucket \
  --log-bucket flow-logs-for-website-lb
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Authorize the DDoS Response Team](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [DisassociateDrtLogBucket AWS CLI Command Reference](#).

## disassociate-drt-role

Il seguente esempio di codice mostra come utilizzare `disassociate-drt-role`.

### AWS CLI

Per rimuovere l'autorizzazione DRT al fine di mitigare potenziali attacchi per conto dell'utente

L'`disassociate-drt-role` esempio seguente rimuove l'associazione tra il DRT e l'account. Dopo questa chiamata, non DRT possono più accedere o gestire il tuo account.

```
aws shield disassociate-drt-role
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Authorize the DDoS Response Team](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [DisassociateDrtRole AWS CLI Command Reference](#).

## get-subscription-state

Il seguente esempio di codice mostra come utilizzare `get-subscription-state`.

### AWS CLI

Per recuperare lo stato attuale dell'abbonamento AWS Shield Advanced dell'account

L'esempio seguente recupera lo stato della protezione Shield Advanced per l'account.

```
aws shield get-subscription-state
```

Output:

```
{
  "SubscriptionState": "ACTIVE"
}
```

Per ulteriori informazioni, consulta [How AWS Shield Works](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [GetSubscriptionState AWS CLI Command Reference](#).

## list-attacks

Il seguente esempio di codice mostra come utilizzare `list-attacks`.

### AWS CLI

Per recuperare i riepiloghi degli attacchi da Shield Advanced AWS

L'esempio seguente recupera i riepiloghi degli attacchi per la AWS CloudFront distribuzione specificata durante il periodo di tempo specificato. La risposta include un attacco IDs che è possibile fornire al `describe-attack` comando per ottenere informazioni dettagliate su un attacco.

```
aws shield list-attacks \
  --resource-arns arn:aws:cloudfront::12345678910:distribution/E1PXM22ZVFA0R \
  --start-time FromInclusive=1529280000,ToExclusive=1529300000
```

Output:

```
{
  "AttackSummaries": [
    {
      "AttackId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/
E1PXM22ZVFAOR",
      "StartTime": 1529280000.0,
      "EndTime": 1529449200.0,
      "AttackVectors": [
        {
          "VectorType": "SYN_FLOOD"
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, consulta Review [DDoSIncidents](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [ListAttacks](#) in AWS CLI Command Reference.

## list-protections

Il seguente esempio di codice mostra come utilizzare `list-protections`.

### AWS CLI

Per recuperare i riepiloghi della protezione da Shield Advanced AWS

L'`list-protections` esempio seguente recupera i riepiloghi delle protezioni abilitate per l'account.

```
aws shield list-protections
```

Output:

```
{
  "Protections": [
    {
      "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "Name": "Protection for CloudFront distribution",

```

```
        "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/
E198WC25FX0WY8"
    }
]
}
```

Per ulteriori informazioni, consulta [Specificare le risorse da proteggere](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [ListProtections AWS CLI](#) Command Reference.

## update-emergency-contact-settings

Il seguente esempio di codice mostra come utilizzare `update-emergency-contact-settings`.

### AWS CLI

Per definire gli indirizzi e-mail di emergenza archiviati presso il DRT

L'esempio seguente definisce due indirizzi di posta elettronica che DRT devono contattare in caso di risposta a un attacco sospetto.

```
aws shield update-emergency-contact-settings \
    --emergency-contact-list EmailAddress=ops@example.com EmailAddress=ddos-
notifications@example.com
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [How AWS Shield Works](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [UpdateEmergencyContactSettings AWS CLI](#) Command Reference.

## update-subscription

Il seguente esempio di codice mostra come utilizzare `update-subscription`.

### AWS CLI

Per modificare l'abbonamento AWS Shield Advanced dell'account

L'esempio seguente abilita il rinnovo automatico dell'abbonamento AWS Shield Advanced per l'account.

```
aws shield update-subscription \  
  --auto-renew ENABLED
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [How AWS Shield Works](#) nella AWS Shield Advanced Developer Guide.

- Per API i dettagli, consulta [UpdateSubscription AWS CLI Command Reference](#).

## Esempi di firmatari che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Signer.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **cancel-signing-profile**

Il seguente esempio di codice mostra come utilizzare `cancel-signing-profile`.

AWS CLI

Per eliminare un profilo di firma

L'esempio seguente rimuove un profilo di firma esistente da AWS Signer.

```
aws signer cancel-signing-profile \  
  --profile-name MyProfile1
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [CancelSigningProfile AWS CLI Command Reference](#).

## describe-signing-job

Il seguente esempio di codice mostra come utilizzare `describe-signing-job`.

### AWS CLI

Per visualizzare i dettagli su un processo di firma

L'esempio seguente visualizza i dettagli sul processo di firma specificato.

```
aws signer describe-signing-job \
  --job-id 2065c468-73e2-4385-a6c9-0123456789abc
```

Output:

```
{
  "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
}
```

- Per API i dettagli, vedere [DescribeSigningJob](#) in AWS CLI Command Reference.

## get-signing-platform

Il seguente esempio di codice mostra come utilizzare `get-signing-platform`.

### AWS CLI

Per visualizzare i dettagli su una piattaforma di firma

L'`get-signing-platform`esempio seguente visualizza i dettagli sulla piattaforma di firma specificata.

```
aws signer get-signing-platform \
  --platform-id AmazonFreeRTOS-TI-CC3220SF
```

Output:

```
{
  "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"
      ]
    }
  }
}

```

- Per API i dettagli, vedere [GetSigningPlatform](#) in AWS CLI Command Reference.

## get-signing-profile

Il seguente esempio di codice mostra come utilizzare `get-signing-profile`.

### AWS CLI

Per visualizzare i dettagli su un profilo di firma

L'`get-signing-profile` esempio seguente visualizza i dettagli sul profilo di firma specificato.

```

aws signer get-signing-profile \
  --profile-name MyProfile3

```

Output:

```

{
  "platformId": "AmazonFreeRTOS-TI-CC3220SF",
  "profileName": "MyProfile3",
  "status": "Active",
  "signingMaterial": {
    "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
  }
}

```

- Per API i dettagli, vedere [GetSigningProfile](#) in AWS CLI Command Reference.

## list-signing-jobs

Il seguente esempio di codice mostra come utilizzare `list-signing-jobs`.

## AWS CLI

Per elencare tutti i lavori di firma

L'`list-signing-jobs` seguente mostra i dettagli su tutti i lavori di firma per l'account.

```
aws signer list-signing-jobs
```

In questo esempio, vengono restituiti due processi, uno riuscito e uno fallito.

```
{
  "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"
  }
]
}
```

- Per API i dettagli, vedere [ListSigningJobs](#) in AWS CLI Command Reference.

## list-signing-platforms

Il seguente esempio di codice mostra come utilizzare `list-signing-platforms`.

### AWS CLI

Per elencare tutte le piattaforme di firma

L'`list-signing-platforms` esempio seguente mostra i dettagli su tutte le piattaforme di firma disponibili.

```
aws signer list-signing-platforms
```

Output:

```
{
  "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"
      ]
    }
  }
]
}

```

- Per API i dettagli, vedere [ListSigningPlatforms](#) in AWS CLI Command Reference.

## list-signing-profiles

Il seguente esempio di codice mostra come utilizzare `list-signing-profiles`.

### AWS CLI

Per elencare tutti i profili di firma

L'`list-signing-profiles` esempio seguente mostra i dettagli su tutti i profili di firma per l'account.

```
aws signer list-signing-profiles
```

Output:

```
{
  "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"
      }
    }
  ]
}
```

- Per API i dettagli, vedere [ListSigningProfiles](#) in AWS CLI Command Reference.

## put-signing-profile

Il seguente esempio di codice mostra come utilizzare `put-signing-profile`.

AWS CLI

Per creare un profilo di firma

L'`put-signing-profile` esempio seguente crea un profilo di firma utilizzando il certificato e la piattaforma specificati.

```
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
```

Output:

```
{  
  "arn": "arn:aws:signer:us-west-2:123456789012:/signing-profiles/MyProfile6"  
}
```

- Per API i dettagli, vedere [PutSigningProfile](#) in AWS CLI Command Reference.

## start-signing-job

Il seguente esempio di codice mostra come utilizzare `start-signing-job`.

AWS CLI

Per iniziare un processo di firma

L'`start-signing-job` esempio seguente avvia un processo di firma sul codice trovato nella fonte specificata. Utilizza il profilo specificato per eseguire la firma e inserisce il codice firmato nella destinazione specificata.

```
aws signer start-signing-job \  
  --source 's3={bucketName=signer-source,key=MyCode.rb,version=PMyFaUTgsQh5ZdMCcoCe6pT1g0pgB_M4}' \  
  --destination 's3={bucketName=signer-destination,prefix=signed-}' \  
  --profile-name MyProfile7
```

L'output è l'ID del processo di firma.

```
{  
  "jobId": "2065c468-73e2-4385-a6c9-0123456789abc"  
}
```

- Per API i dettagli, vedere [StartSigningJob](#) in AWS CLI Command Reference.



## Esempi di utilizzo di Snowball AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Snowball.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni illustrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **get-snowball-usage**

Il seguente esempio di codice mostra come utilizzare `get-snowball-usage`.

AWS CLI

Per ottenere informazioni sul limite del servizio Snowball per il tuo account

L'`get-snowball-usage` esempio seguente mostra informazioni sul limite del servizio Snowball per il tuo account e anche sul numero di Snowball che il tuo account ha in uso.

```
aws snowball get-snowball-usage
```

Output:

```
{
  "SnowballLimit": 1,
  "SnowballsInUse": 0
}
```

FORper ulteriori informazioni, consulta [AWS Snowball Edge Limits](#) nella AWS Snowball Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetSnowballUsageReference](#).

## list-jobs

Il seguente esempio di codice mostra come utilizzare `list-jobs`.

### AWS CLI

Per elencare le offerte di lavoro attuali in Snowball presenti nel tuo account

L'`list-jobs` esempio seguente mostra una serie di `JobListEntry` oggetti. In questo esempio, viene elencato un singolo lavoro.

```
aws snowball list-jobs
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Jobs for AWS Snowball Edge nella AWS Snowball Developer Guide](#).

- Per API i dettagli, consulta AWS CLI Command [ListJobsReference](#).

## SNSEsempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con AmazonSNS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Gli scenari sono esempi di codice che mostrano come eseguire attività specifiche richiamando più funzioni all'interno di un servizio o combinandole con altre Servizi AWS.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)
- [Scenari](#)

## Azioni

### **add-permission**

Il seguente esempio di codice mostra come utilizzare `add-permission`.

#### AWS CLI

Per aggiungere un'autorizzazione a un argomento

L'`add-permission` seguente aggiunge l'autorizzazione per l' AWS account 987654321098 a utilizzare l'azione con l'argomento specificato in AWS account123456789012.

```
aws sns add-permission \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --label Publish-Permission \  
  --aws-account-id 987654321098 \  
  --action-name Publish
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [AddPermission](#) in AWS CLI Command Reference.

### **check-if-phone-number-is-opted-out**

Il seguente esempio di codice mostra come utilizzare `check-if-phone-number-is-opted-out`.

## AWS CLI

Per verificare la disattivazione di un SMS messaggio per un numero di telefono

L'check-if-phone-number-is-opted-outesempio seguente verifica se al numero di telefono specificato è stata disattivata la ricezione di SMS messaggi dall'account corrente AWS .

```
aws sns check-if-phone-number-is-opted-out \  
  --phone-number +1555550100
```

Output:

```
{  
  "isOptedOut": false  
}
```

- Per API i dettagli, vedere [CheckIfPhoneNumberIsOptedOut](#) in AWS CLI Command Reference.

## confirm-subscription

Il seguente esempio di codice mostra come utilizzareconfirm-subscription.

### AWS CLI

Conferma di una sottoscrizione

Il confirm-subscription comando seguente completa il processo di conferma avviato quando l'utente ha sottoscritto un SNS argomento denominato. my-topic Il parametro --token proviene dal messaggio di conferma inviato all'endpoint di notifica specificato nella chiamata subscribe.

```
aws sns confirm-subscription \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \  
  --  
  token 2336412f37fb687f5d51e6e241d7700ae02f7124d8268910b858cb4db727ceeb2474bb937929d3bdd7ce5a
```

Output:

```
{  
  "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-  
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
```

```
}
```

- Per API i dettagli, vedere [ConfirmSubscription](#) in AWS CLI Command Reference.

## create-platform-application

Il seguente esempio di codice mostra come utilizzare `create-platform-application`.

### AWS CLI

Per creare un'applicazione di piattaforma

L'`create-platform-application` esempio seguente crea un'applicazione della piattaforma Google Firebase utilizzando le credenziali di piattaforma specificate.

```
aws sns create-platform-application \  
  --name MyApplication \  
  --platform GCM \  
  --attributes PlatformCredential=EXAMPLEabcd12345jklm67890stuv12345bcdef
```

Output:

```
{  
  "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/GCM/  
  MyApplication"  
}
```

- Per API i dettagli, consulta Command [CreatePlatformApplication](#) Reference AWS CLI .

## create-topic

Il seguente esempio di codice mostra come utilizzare `create-topic`.

### AWS CLI

Per creare un argomento SNS

L'`create-topic` esempio seguente crea un SNS argomento denominato `my-topic`.

```
aws sns create-topic \  
  --name my-topic
```

```
--name my-topic
```

Output:

```
{
  "ResponseMetadata": {
    "RequestId": "1469e8d7-1642-564e-b85d-a19b4b341f83"
  },
  "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
}
```

Per ulteriori informazioni, consulta [Using the AWS Command Line Interface with Amazon SQS e Amazon SNS](#) nella AWS Command Line Interface User Guide.

- Per API i dettagli, consulta [CreateTopic AWS CLI](#) Command Reference.

## delete-endpoint

Il seguente esempio di codice mostra come utilizzare `delete-endpoint`.

AWS CLI

Per eliminare un endpoint dell'applicazione della piattaforma

L'`delete-endpoint` esempio seguente elimina l'endpoint dell'applicazione di piattaforma specificato.

```
aws sns delete-endpoint \
  --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/MyApplication/12345678-abcd-9012-efgh-345678901234
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteEndpoint](#) in AWS CLI Command Reference.

## delete-platform-application

Il seguente esempio di codice mostra come utilizzare `delete-platform-application`.

AWS CLI

Per eliminare un'applicazione della piattaforma

L'`delete-platform-application` seguente elimina l'applicazione di piattaforma specificata.

```
aws sns delete-platform-application \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/ADM/MyApplication
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeletePlatformApplication](#) in AWS CLI Command Reference.

## **delete-topic**

Il seguente esempio di codice mostra come utilizzare `delete-topic`.

### AWS CLI

Per eliminare un SNS argomento

L'`delete-topic` seguente elimina l'SNS argomento specificato.

```
aws sns delete-topic \  
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteTopic](#) in AWS CLI Command Reference.

## **get-endpoint-attributes**

Il seguente esempio di codice mostra come utilizzare `get-endpoint-attributes`.

### AWS CLI

Per elencare gli attributi degli endpoint dell'applicazione della piattaforma

L'`get-endpoint-attributes` seguente elenca gli attributi per l'endpoint dell'applicazione di piattaforma specificato.

```
aws sns get-endpoint-attributes \  
  --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/MyApplication/12345678-abcd-9012-efgh-345678901234
```

Output:

```
{
  "Attributes": {
    "Enabled": "true",
    "Token": "EXAMPLE12345..."
  }
}
```

- Per API i dettagli, vedere [GetEndpointAttributes](#) in AWS CLI Command Reference.

## get-platform-application-attributes

Il seguente esempio di codice mostra come utilizzare `get-platform-application-attributes`.

AWS CLI

Per elencare gli attributi dell'applicazione della piattaforma

L'`get-platform-application-attributes` esempio seguente elenca gli attributi per l'applicazione di piattaforma specificata.

```
aws sns get-platform-application-attributes \
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/MPNS/
MyApplication
```

Output:

```
{
  "Attributes": {
    "Enabled": "true",
    "SuccessFeedbackSampleRate": "100"
  }
}
```

- Per API i dettagli, vedere [GetPlatformApplicationAttributes](#) in AWS CLI Command Reference.

## get-sms-attributes

Il seguente esempio di codice mostra come utilizzare `get-sms-attributes`.



## AWS CLI

Per elencare gli attributi predefiniti dei SMS messaggi

L'`get-sms-attributes` seguente elenca gli attributi predefiniti per l'invio SMS di messaggi.

```
aws sns get-sms-attributes
```

Output:

```
{
  "attributes": {
    "DefaultSenderId": "MyName"
  }
}
```

- Per API i dettagli, vedere [GetSMSAttributes](#) in AWS CLI Command Reference.

## get-subscription-attributes

Il seguente esempio di codice mostra come usare `get-subscription-attributes`.

## AWS CLI

Per recuperare gli attributi di sottoscrizione per un argomento

Di seguito `get-subscription-attributes` vengono visualizzati gli attributi della sottoscrizione specificata. È possibile ottenere il `subscription-arn` dall'output del `list-subscriptions` comando.

```
aws sns get-subscription-attributes \
  --subscription-arn "arn:aws:sns:us-west-2:123456789012:my-
  topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
```

Output:

```
{
  "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"
  }
}

```

- Per API i dettagli, vedere [GetSubscriptionAttributes](#) in AWS CLI Command Reference.

## get-topic-attributes

Il seguente esempio di codice mostra come utilizzare `get-topic-attributes`.

### AWS CLI

Recupero degli attributi di un argomento

Nell'esempio `get-topic-attributes` seguente vengono visualizzati gli attributi per l'argomento specificato.

```

aws sns get-topic-attributes \
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"

```

Output:

```

{
  "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\"
}
}

```

- Per API i dettagli, vedere [GetTopicAttributes](#) in AWS CLI Command Reference.

## list-endpoints-by-platform-application

Il seguente esempio di codice mostra come utilizzare `list-endpoints-by-platform-application`.

### AWS CLI

Per elencare gli endpoint di un'applicazione di piattaforma

L'`list-endpoints-by-platform-application` esempio seguente elenca gli endpoint e gli attributi degli endpoint per l'applicazione di piattaforma specificata.

```

aws sns list-endpoints-by-platform-application \
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication

```

Output:

```

{
  "Endpoints": [
    {
      "Attributes": {
        "Token": "EXAMPLE12345...",
        "Enabled": "true"
      },
      "EndpointArn": "arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234"
    }
  ]
}

```

- Per API i dettagli, vedere [ListEndpointsByPlatformApplication](#) in AWS CLI Command Reference.

## **list-phone-numbers-opted-out**

Il seguente esempio di codice mostra come utilizzare `list-phone-numbers-opted-out`.

### AWS CLI

Per elencare le opzioni di disattivazione dei SMS messaggi

L'`list-phone-numbers-opted-out` esempio seguente elenca i numeri di telefono a cui è stato negato il consenso alla ricezione dei messaggi. SMS

```
aws sns list-phone-numbers-opted-out
```

Output:

```
{
  "phoneNumbers": [
    "+15555550100"
  ]
}
```

- Per API i dettagli, vedere [ListPhoneNumbersOptedOut](#) in AWS CLI Command Reference.

## **list-platform-applications**

Il seguente esempio di codice mostra come utilizzare `list-platform-applications`.

### AWS CLI

Per elencare le applicazioni della piattaforma

L'`list-platform-applications` esempio seguente elenca le applicazioni della piattaforma per ADM e MPNS.

```
aws sns list-platform-applications
```

Output:

```
{
  "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"
      }
    }
  ]
}
```

- Per API i dettagli, vedere [ListPlatformApplications](#) in AWS CLI Command Reference.

## list-subscriptions-by-topic

Il seguente esempio di codice mostra come utilizzare `list-subscriptions-by-topic`.

### AWS CLI

Per elencare le sottoscrizioni associate a un argomento

Quanto segue `list-subscriptions-by-topic` recupera un elenco di SNS sottoscrizioni associate all'argomento specificato.

```
aws sns list-subscriptions-by-topic \
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

Output:

```
{
  "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"
}
```

- Per API i dettagli, vedere [ListSubscriptionsByTopic](#) in AWS CLI Command Reference.

## list-subscriptions

Il seguente esempio di codice mostra come utilizzare `list-subscriptions`.

### AWS CLI

Per elencare i tuoi SNS abbonamenti

L'esempio seguente mostra un elenco degli SNS abbonamenti presenti nel tuo AWS account.

```
aws sns list-subscriptions
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ListSubscriptions](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di un argomento

L'`list-tags-for-resource` esempio seguente elenca i tag per l'SNS argomento Amazon specificato.

```
aws sns list-tags-for-resource \
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic
```

Output:

```
{
  "Tags": [
    {
      "Key": "Team",
      "Value": "Alpha"
    }
  ]
}
```

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## list-topics

Il seguente esempio di codice mostra come utilizzare `list-topics`.

### AWS CLI

Per elencare i tuoi SNS argomenti

L'`list-topics` esempio seguente elenca tutti SNS gli argomenti del tuo AWS account.

```
aws sns list-topics
```

Output:

```
{
  "Topics": [
    {
      "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
    }
  ]
}
```

- Per API i dettagli, vedere [ListTopics](#) in AWS CLI Command Reference.

## opt-in-phone-number

Il seguente esempio di codice mostra come utilizzare `opt-in-phone-number`.

### AWS CLI

Per attivare la ricezione dei messaggi SMS

L'`opt-in-phone-number` esempio seguente imposta il numero di telefono specificato per la ricezione SMS dei messaggi.

```
aws sns opt-in-phone-number \
  --phone-number +15555550100
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [OptInPhoneNumber](#) in AWS CLI Command Reference.

## publish

Il seguente esempio di codice mostra come utilizzare `publish`.

### AWS CLI

Esempio 1: pubblicazione di un messaggio in un argomento

L'`publish` esempio seguente pubblica il messaggio specificato SNS sull'argomento specificato. Il messaggio proviene da un file di testo che consente di includere interruzioni di riga.

```
aws sns publish \
  --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic" \
```



```
--message file://message.txt
```

Contenuto di message.txt.

```
Hello World  
Second Line
```

Output:

```
{  
  "MessageId": "123a45b6-7890-12c3-45d6-111122223333"  
}
```

Esempio 2: pubblicare un SMS messaggio su un numero di telefono

Nell'esempio publish seguente viene pubblicato il messaggio Hello world! sul numero di telefono+1-555-555-0100.

```
aws sns publish \  
  --message "Hello world!" \  
  --phone-number +1-555-555-0100
```

Output:

```
{  
  "MessageId": "123a45b6-7890-12c3-45d6-333322221111"  
}
```

- Per API i dettagli, consulta [Pubblica](#) in AWS CLI Command Reference.

## put-data-protection-policy

Il seguente esempio di codice mostra come utilizzareput-data-protection-policy.

AWS CLI

Per impostare la politica di protezione dei dati

Esempio 1: impedire agli editori di pubblicare messaggi con CreditCardNumber

L'put-data-protection-policyesempio seguente impedisce agli editori di pubblicare messaggi con. CreditCardNumber

```
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\": {}}}]}'
```

Questo comando non produce alcun output.

Esempio 2: per caricare i parametri da un file

Quanto segue put-data-protection-policy carica i parametri da un file.

```
aws sns put-data-protection-policy \
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
  --data-protection-policy file://policy.json
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutDataProtectionPolicy](#) in AWS CLI Command Reference.

## remove-permission

Il seguente esempio di codice mostra come utilizzare remove-permission.

### AWS CLI

Per rimuovere un'autorizzazione da un argomento

L'remove-permissionesempio seguente rimuove l'autorizzazione Publish-Permission dall'argomento specificato.

```
aws sns remove-permission \
  --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
  --label Publish-Permission
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [RemovePermission](#) in AWS CLI Command Reference.

## set-endpoint-attributes

Il seguente esempio di codice mostra come utilizzare `set-endpoint-attributes`.

### AWS CLI

Per impostare gli attributi degli endpoint

L'esempio seguente disabilita l'endpoint dell'applicazione della piattaforma specificato.

```
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
```

Output:

```
{  
  "Attributes": {  
    "Enabled": "false",  
    "Token": "EXAMPLE12345..."  
  }  
}
```

- Per API i dettagli, vedere [SetEndpointAttributes](#) in AWS CLI Command Reference.

## set-platform-application-attributes

Il seguente esempio di codice mostra come utilizzare `set-platform-application-attributes`.

### AWS CLI

Per impostare gli attributi dell'applicazione della piattaforma

L'esempio seguente imposta l'attributo `EventDeliveryFailure` per l'applicazione ARN di piattaforma specificata sull'argomento Amazon specificato.

```
aws sns set-platform-application-attributes \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/MyApplication \  
  --attributes EventDeliveryFailure=true
```

```
--attributes EventDeliveryFailure=arn:aws:sns:us-west-2:123456789012:AnotherTopic
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [SetPlatformApplicationAttributes AWS CLI Command Reference](#).

## set-sms-attributes

Il seguente esempio di codice mostra come utilizzare `set-sms-attributes`.

### AWS CLI

Per impostare gli attributi dei SMS messaggi

L'`set-sms-attributes` esempio seguente imposta l'ID mittente predefinito per SMS MyName i messaggi su.

```
aws sns set-sms-attributes \  
  --attributes DefaultSenderId=MyName
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [SetSMSAttributes](#) in AWS CLI Command Reference.

## set-subscription-attributes

Il seguente esempio di codice mostra come utilizzare `set-subscription-attributes`.

### AWS CLI

Impostazione degli attributi della sottoscrizione

L'`set-subscription-attributes` esempio seguente imposta l'`RawMessageDelivery` attributo su un SQS abbonamento.

```
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
```

Questo comando non produce alcun output.

L'esempio seguente imposta un `FilterPolicy` attributo a un SQS abbonamento.

```
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\"] }"
```

Questo comando non produce alcun output.

L'esempio seguente rimuove l'`FilterPolicy` attributo da una SQS sottoscrizione.

```
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 "{}"
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [SetSubscriptionAttributes](#) in AWS CLI Command Reference.

## set-topic-attributes

Il seguente esempio di codice mostra come utilizzare `set-topic-attributes`.

### AWS CLI

Impostazione di un attributo per un argomento

Nell'esempio `set-topic-attributes` seguente vengono impostati gli attributi `DisplayName` per l'argomento specificato.

```
aws sns set-topic-attributes \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --attribute-name DisplayName \  
  --attribute-value MyTopicDisplayName
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [SetTopicAttributes](#) in AWS CLI Command Reference.

## subscribe

Il seguente esempio di codice mostra come utilizzare `subscribe`.

### AWS CLI

Effettuare la sottoscrizione a un argomento

Attraverso il comando `subscribe` seguente viene effettuata la sottoscrizione all'argomento specificato utilizzando un indirizzo e-mail.

```
aws sns subscribe \  
  --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \  
  --protocol email \  
  --notification-endpoint my-email@example.com
```

Output:

```
{  
  "SubscriptionArn": "pending confirmation"  
}
```

- Per API i dettagli, consulta [Subscribe](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Aggiungere un tag a un argomento

L'`tag-resource` seguente aggiunge un tag di metadati all'SNS argomento Amazon specificato.

```
aws sns tag-resource \  
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --tag-key MyTagKey \  
  --tag-value MyTagValue
```

```
--tags Key=Team, Value=Alpha
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [TagResource AWS CLI Command Reference](#).

## unsubscribe

Il seguente esempio di codice mostra come utilizzare `unsubscribe`.

### AWS CLI

Annullamento della sottoscrizione a un argomento

Nell'esempio `unsubscribe` seguente viene eliminata la sottoscrizione specificata a un argomento.

```
aws sns unsubscribe \  
  --subscription-arn arn:aws:sns:us-west-2:0123456789012:my-  
topic:8a21d249-4329-4871-acc6-7be709c6ea7f
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [Unsubscribe](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere un tag da un argomento

L'`untag-resource` esempio seguente rimuove tutti i tag con le chiavi specificate dall'`SNS` argomento Amazon specificato.

```
aws sns untag-resource \  
  --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \  
  --tag-keys Team
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [UntagResource AWS CLI Command Reference](#).

## Scenari

### Creazione di un endpoint di piattaforma per notifiche push

Il seguente esempio di codice mostra come creare un endpoint della piattaforma per le notifiche SNS push di Amazon.

#### AWS CLI

Creazione di un endpoint dell'applicazione della piattaforma

Nell'esempio `create-platform-endpoint` seguente viene creato un endpoint per l'applicazione della piattaforma indicata utilizzando il token specificato.

```
aws sns create-platform-endpoint \  
  --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/  
MyApplication \  
  --token EXAMPLE12345...
```

Output:

```
{  
  "EndpointArn": "arn:aws:sns:us-west-2:1234567890:endpoint/GCM/  
MyApplication/12345678-abcd-9012-efgh-345678901234"  
}
```

## SQSEsempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con AmazonSQS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

#### Argomenti

- [Azioni](#)



## Azioni

### **add-permission**

Il seguente esempio di codice mostra come utilizzare `add-permission`.

#### AWS CLI

Per aggiungere un'autorizzazione a una coda

Questo esempio consente all' AWS account specificato di inviare messaggi alla coda specificata.

Comando:

```
aws sqs add-permission --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --label SendMessageFromMyQueue --aws-account-ids 12345EXAMPLE --actions SendMessage
```

Output:

```
None.
```

- Per API i dettagli, vedere [AddPermission](#) in AWS CLI Command Reference.

### **cancel-message-move-task**

Il seguente esempio di codice mostra come utilizzare `cancel-message-move-task`.

#### AWS CLI

Per annullare un messaggio, sposta un'operazione.

L'`cancel-message-move-task` esempio seguente annulla l'attività di spostamento del messaggio specificata.

```
aws sqs cancel-message-move-task \  
  --task-handle AQEB6nR4...Hz1vZQ==
```

Output:

```
{  
  "ApproximateNumberOfMessagesMoved": 102
```

```
}
```

Per ulteriori informazioni, consulta [Amazon SQS API permissions: Actions and resource reference](#) nella Developer Guide.

- Per API i dettagli, consulta [CancelMessageMoveTask AWS CLI Command Reference](#).

## change-message-visibility-batch

Il seguente esempio di codice mostra come utilizzare `change-message-visibility-batch`.

### AWS CLI

Per modificare la visibilità del timeout di più messaggi come batch

Questo esempio modifica la visibilità del timeout dei 2 messaggi specificati su 10 ore (10 ore\* 60 minuti\* 60 secondi).

Comando:

```
aws sqs change-message-visibility-batch --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://change-message-visibility-batch.json
```

File di input (.json): `change-message-visibility-batch`

```
[
  {
    "Id": "FirstMessage",
    "ReceiptHandle": "AQEBhz2q...Jf3kaw==",
    "VisibilityTimeout": 36000
  },
  {
    "Id": "SecondMessage",
    "ReceiptHandle": "AQEBkTUH...HifSnw==",
    "VisibilityTimeout": 36000
  }
]
```

Output:

```
{
```

```
"Successful": [
  {
    "Id": "SecondMessage"
  },
  {
    "Id": "FirstMessage"
  }
]
```

- Per API i dettagli, vedere [ChangeMessageVisibilityBatch](#) in AWS CLI Command Reference.

## change-message-visibility

Il seguente esempio di codice mostra come utilizzare `change-message-visibility`.

### AWS CLI

Per modificare la visibilità del timeout di un messaggio

Questo esempio modifica la visibilità del timeout del messaggio specificato su 10 ore (10 ore\* 60 minuti\* 60 secondi).

Comando:

```
aws sqs change-message-visibility --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --receipt-handle AQEBTpyI...t6HyQg== --visibility-timeout 36000
```

Output:

```
None.
```

- Per API i dettagli, vedere [ChangeMessageVisibility](#) in AWS CLI Command Reference.

## create-queue

Il seguente esempio di codice mostra come utilizzare `create-queue`.

### AWS CLI

Per creare una coda

Questo esempio crea una coda con il nome specificato, imposta il periodo di conservazione dei messaggi su 3 giorni (3 giorni\* 24 ore\* 60 minuti \* 60 secondi) e imposta la coda di lettere morte della coda sulla coda specificata con un numero massimo di ricezione di 1.000 messaggi.

Comando:

```
aws sqs create-queue --queue-name MyQueue --attributes file://create-queue.json
```

File di input (create-queue.json):

```
{
  "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-east-1:80398EXAMPLE:MyDeadLetterQueue\", \"maxReceiveCount\": \"1000\"}\",
  "MessageRetentionPeriod": "259200"
}
```

Output:

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

- Per i API dettagli, vedere in Command Reference. [CreateQueue](#) AWS CLI

## delete-message-batch

Il seguente esempio di codice mostra come utilizzare `delete-message-batch`.

AWS CLI

Per eliminare più messaggi in batch

Questo esempio elimina i messaggi specificati.

Comando:

```
aws sqs delete-message-batch --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://delete-message-batch.json
```

File di input (delete-message-batch.json):

```
[
  {
    "Id": "FirstMessage",
    "ReceiptHandle": "AQEB1mg1...Z4GuLw=="
  },
  {
    "Id": "SecondMessage",
    "ReceiptHandle": "AQEBLsYM...VQubAA=="
  }
]
```

Output:

```
{
  "Successful": [
    {
      "Id": "FirstMessage"
    },
    {
      "Id": "SecondMessage"
    }
  ]
}
```

- Per API i dettagli, vedere [DeleteMessageBatch](#) in AWS CLI Command Reference.

## delete-message

Il seguente esempio di codice mostra come utilizzare `delete-message`.

### AWS CLI

Per eliminare un messaggio

Questo esempio elimina il messaggio specificato.

Comando:

```
aws sqs delete-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --receipt-handle AQEBRXTo...q2doVA==
```

Output:

```
None.
```

- Per API i dettagli, vedere [DeleteMessage](#) in AWS CLI Command Reference.

## delete-queue

Il seguente esempio di codice mostra come utilizzare `delete-queue`.

### AWS CLI

Per eliminare una coda

Questo esempio elimina la coda specificata.

Comando:

```
aws sqs delete-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewerQueue
```

Output:

```
None.
```

- Per API i dettagli, vedere [DeleteQueue](#) in AWS CLI Command Reference.

## get-queue-attributes

Il seguente esempio di codice mostra come utilizzare `get-queue-attributes`.

### AWS CLI

Per ottenere gli attributi di una coda

Questo esempio ottiene tutti gli attributi della coda specificata.

Comando:

```
aws sqs get-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names All
```

Output:

```
{
  "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"
  }
}
```

Questo esempio ottiene solo la dimensione massima dei messaggi e gli attributi di timeout di visibilità della coda specificati.

Comando:

```
aws sqs get-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attribute-names MaximumMessageSize VisibilityTimeout
```

Output:

```
{
  "Attributes": {
    "VisibilityTimeout": "30",
    "MaximumMessageSize": "262144"
  }
}
```

- Per API i dettagli, vedere [GetQueueAttributes](#) in AWS CLI Command Reference.

## get-queue-url

Il seguente esempio di codice mostra come utilizzare `get-queue-url`.

## AWS CLI

Per creare una coda URL

Questo esempio ottiene le code specificate. URL

Comando:

```
aws sqs get-queue-url --queue-name MyQueue
```

Output:

```
{  
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"  
}
```

- Per API i dettagli, vedere [GetQueueUrl](#) in AWS CLI Command Reference.

## list-dead-letter-source-queues

Il seguente esempio di codice mostra come utilizzare `list-dead-letter-source-queues`.

## AWS CLI

Per elencare le code di sorgenti con lettere morte

Questo esempio elenca le code associate alla coda di origine delle lettere morte specificata.

Comando:

```
aws sqs list-dead-letter-source-queues --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue
```

Output:

```
{  
  "queueUrls": [  
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",  
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"  
  ]  
}
```



- Per API i dettagli, vedere [ListDeadLetterSourceQueues](#) in AWS CLI Command Reference.

## list-message-move-tasks

Il seguente esempio di codice mostra come utilizzare `list-message-move-tasks`.

### AWS CLI

Per elencare il messaggio, sposta le attività.

L'`list-message-move-tasks` esempio seguente elenca le 2 attività di spostamento dei messaggi più recenti nella coda specificata.

```
aws sqs list-message-move-tasks \  
  --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue \  
  --max-results 2
```

### Output:

```
{  
  "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  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Amazon SQS API permissions: Actions and resource reference](#) nella Developer Guide.

- Per API i dettagli, consulta [ListMessageMoveTasks AWS CLI Command Reference](#).

## list-queue-tags

Il seguente esempio di codice mostra come utilizzare `list-queue-tags`.

### AWS CLI

Per elencare tutti i tag di allocazione dei costi per una coda

L'esempio seguente visualizza tutti i tag di allocazione dei costi associati alla coda specificata.

```
aws sqs list-queue-tags \  
  --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue
```

Output:

```
{  
  "Tags": {  
    "Team": "Alpha"  
  }  
}
```

Per ulteriori informazioni, consulta [Elencare i tag di allocazione dei costi](#) nella Guida per gli sviluppatori di Amazon Simple Queue Service.

- Per API i dettagli, consulta AWS CLI Command [ListQueueTagsReference](#).

## list-queues

Il seguente esempio di codice mostra come utilizzare `list-queues`.

### AWS CLI

Per elencare le code

Questo esempio elenca tutte le code.

Comando:

```
aws sqs list-queues
```

Output:

```
{
  "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"
  ]
}
```

Questo esempio elenca solo le code che iniziano con «My».

Comando:

```
aws sqs list-queues --queue-name-prefix My
```

Output:

```
{
  "QueueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"
  ]
}
```

- Per API i dettagli, vedere [ListQueues](#) in AWS CLI Command Reference.

## purge-queue

Il seguente esempio di codice mostra come utilizzare `purge-queue`.

AWS CLI

Per eliminare una coda

Questo esempio elimina tutti i messaggi nella coda specificata.

Comando:

```
aws sqs purge-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue
```

Output:

```
None.
```

- Per API i dettagli, vedere [PurgeQueue](#) in AWS CLI Command Reference.

## receive-message

Il seguente esempio di codice mostra come utilizzare `receive-message`.

AWS CLI

Per ricevere un messaggio

Questo esempio riceve fino a 10 messaggi disponibili, restituendo tutti gli attributi disponibili.

Comando:

```
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
```

Output:

```
{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEBzbVv...fqNzFw==",
      "MD5ofBody": "1000f835...a35411fa",
      "MD5ofMessageAttributes": "9424c491...26bc3ae7",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "ApproximateFirstReceiveTimestamp": "1442428276921",
        "SenderId": "AIDAIASZKMSNQ7TEXAMPLE",
```

```

    "ApproximateReceiveCount": "5",
    "SentTimestamp": "1442428276921"
  },
  "MessageAttributes": {
    "PostalCode": {
      "DataType": "String",
      "StringValue": "ABC123"
    },
    "City": {
      "DataType": "String",
      "StringValue": "Any City"
    }
  }
}
]
}

```

Questo esempio riceve il prossimo messaggio disponibile, restituendo solo gli `SentTimestamp` attributi `SenderId` e e l'attributo `PostalCode` message.

Comando:

```

aws sqs receive-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names SenderId SentTimestamp --message-attribute-names PostalCode

```

Output:

```

{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEB6nR4...HzlvZQ==",
      "MD5ofBody": "1000f835...a35411fa",
      "MD5ofMessageAttributes": "b8e89563...e088e74f",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "SenderId": "AIDAIKMSNQ7TEXAMPLE",
        "SentTimestamp": "1442428276921"
      },
      "MessageAttributes": {
        "PostalCode": {
          "DataType": "String",

```

```
        "StringValue": "ABC123"
      }
    }
  ]
}
```

- Per API i dettagli, vedere [ReceiveMessage](#) in AWS CLI Command Reference.

## remove-permission

Il seguente esempio di codice mostra come utilizzare `remove-permission`.

### AWS CLI

Per rimuovere un'autorizzazione

Questo esempio rimuove l'autorizzazione con l'etichetta specificata dalla coda specificata.

Comando:

```
aws sqs remove-permission --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --label SendMessageFromMyQueue
```

Output:

```
None.
```

- Per API i dettagli, vedere [RemovePermission](#) in AWS CLI Command Reference.

## send-message-batch

Il seguente esempio di codice mostra come utilizzare `send-message-batch`.

### AWS CLI

Per inviare più messaggi in batch

Questo esempio invia 2 messaggi con i corpi dei messaggi, i periodi di ritardo e gli attributi dei messaggi specificati, alla coda specificata.

Comando:

```
aws sqs send-message-batch --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://send-message-batch.json
```

File di input (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"
  }
}
]

```

### Output:

```

{
  "Successful": [
    {
      "MD5ofMessageBody": "203c4a38...7943237e",
      "MD5ofMessageAttributes": "10809b55...baf283ef",
      "Id": "FuelReport-0001-2015-09-16T140731Z",
      "MessageId": "d175070c-d6b8-4101-861d-adeb3EXAMPLE"
    },
    {
      "MD5ofMessageBody": "2cf0159a...c1980595",
      "MD5ofMessageAttributes": "55623928...ae354a25",
      "Id": "FuelReport-0002-2015-09-16T140930Z",
      "MessageId": "f9b7d55d-0570-413e-b9c5-a9264EXAMPLE"
    }
  ]
}

```

- Per API i dettagli, vedere [SendMessageBatch](#) in AWS CLI Command Reference.



## send-message

Il seguente esempio di codice mostra come utilizzare `send-message`.

### AWS CLI

Per inviare un messaggio

Questo esempio invia un messaggio con il corpo del messaggio, il periodo di ritardo e gli attributi del messaggio specificati alla coda specificata.

Comando:

```
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
```

File di input (send-message.json):

```
{
  "City": {
    "DataType": "String",
    "StringValue": "Any City"
  },
  "Greeting": {
    "DataType": "Binary",
    "BinaryValue": "Hello, World!"
  },
  "Population": {
    "DataType": "Number",
    "StringValue": "1250800"
  }
}
```

Output:

```
{
  "MD5ofMessageBody": "51b0a325...39163aa0",
  "MD5ofMessageAttributes": "00484c68...59e48f06",
  "MessageId": "da68f62c-0c07-4bee-bf5f-7e856EXAMPLE"
}
```

- Per API i dettagli, vedere in Command Reference. [SendMessage](#) AWS CLI

## set-queue-attributes

Il seguente esempio di codice mostra come utilizzare `set-queue-attributes`.

### AWS CLI

Per impostare gli attributi della coda

Questo esempio imposta la coda specificata su un ritardo di recapito di 10 secondi, una dimensione massima del messaggio di 128 KB (128 KB \* 1.024 byte), un periodo di conservazione dei messaggi di 3 giorni (3 giorni \* 24 ore \* 60 minuti \* 60 secondi), un tempo di attesa per la ricezione dei messaggi di 20 secondi e un timeout di visibilità predefinito di 60 secondi. Questo esempio associa inoltre la coda di lettere morte specificata a un numero massimo di ricezione di 1.000 messaggi.

Comando:

```
aws sqs set-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attributes file://set-queue-attributes.json
```

File di input (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"
}
```

Output:

```
None.
```

- Per API i dettagli, vedere [SetQueueAttributes](#) in AWS CLI Command Reference.

## start-message-move-task

Il seguente esempio di codice mostra come utilizzare `start-message-move-task`.

### AWS CLI

Esempio 1: \*Per iniziare un messaggio, sposta l'attività\*

L'esempio seguente avvia un'operazione di spostamento dei messaggi per reindirizzare i messaggi dalla coda di lettere non scritte specificata alla coda di origine.

```
aws sqs start-message-move-task \  
  --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue
```

Output:

```
{  
  "TaskHandle": "AQEB6nR4...Hz1vZQ=="  
}
```

Per ulteriori informazioni, consulta [Questo è il titolo dell'argomento nel Nome](#) della guida.

Esempio 2: \*Per iniziare un messaggio, sposta un'attività con una frequenza massima\*

L'esempio seguente avvia un'attività di spostamento dei messaggi per reindirizzare i messaggi dalla coda di lettere morte specificata alla coda di destinazione specificata a una velocità massima di 50 messaggi al secondo.

```
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
```

Output:

```
{  
  "TaskHandle": "AQEB6nR4...Hz1vZQ=="  
}
```

Per ulteriori informazioni, consulta [Amazon SQS API permissions: Actions and resource reference](#) nella Developer Guide.

- Per API i dettagli, consulta [StartMessageMoveTask AWS CLI Command Reference](#).

## tag-queue

Il seguente esempio di codice mostra come utilizzare `tag-queue`.

### AWS CLI

Per aggiungere tag di allocazione dei costi a una coda

L'`tag-queue` esempio seguente aggiunge un tag di allocazione dei costi alla SQS coda Amazon specificata.

```
aws sqs tag-queue \  
  --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \  
  --tags Priority=Highest
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere tag di allocazione dei costi nella Guida](#) per gli sviluppatori di Amazon Simple Queue Service.

- Per API i dettagli, consulta AWS CLI Command [TagQueueReference](#).

## untag-queue

Il seguente esempio di codice mostra come utilizzare `untag-queue`.

### AWS CLI

Per rimuovere i tag di allocazione dei costi da una coda

L'`untag-queue` esempio seguente rimuove un tag di allocazione dei costi dalla SQS coda Amazon specificata.

```
aws sqs untag-queue \  
  --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \  
  --tags Priority=Highest
```

```
--tag-keys "Priority"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Aggiungere tag di allocazione dei costi nella Guida](#) per gli sviluppatori di Amazon Simple Queue Service.

- Per API i dettagli, consulta AWS CLI Command [UntagQueueReference](#).

## Esempi di Storage Gateway utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Storage Gateway.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **describe-gateway-information**

Il seguente esempio di codice mostra come utilizzare `describe-gateway-information`.

#### AWS CLI

Per descrivere un gateway

Il `describe-gateway-information` comando seguente restituisce i metadati relativi al gateway specificato. Per specificare quale gateway descrivere, usa l'Amazon Resource Name (ARN) del gateway nel comando.

Questo esempio specifica un gateway con l'id `sgw-12A3456B` nell'`account123456789012`:

```
aws storagegateway describe-gateway-information --gateway-arn "arn:aws:storagegateway:us-west-2:123456789012:gateway/sgw-12A3456B"
```

Questo comando genera un JSON blocco che contiene metadati sul gateway, ad esempio il nome, le interfacce di rete, il fuso orario configurato e lo stato (indipendentemente dal fatto che il gateway sia in esecuzione o meno).

- Per API i dettagli, vedere [DescribeGatewayInformation](#) in AWS CLI Command Reference.

## list-file-shares

Il seguente esempio di codice mostra come utilizzare `list-file-shares`.

### AWS CLI

Per elencare le condivisioni di file

L'command-name esempio seguente elenca i widget disponibili nel tuo AWS account.

```
aws storagegateway list-file-shares \
  --gateway-arn arn:aws:storagegateway:us-east-1:209870788375:gateway/sgw-FB02E292
```

Output:

```
{
  "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
}
```

Per ulteriori informazioni, vedere [ListFileShares AWS](#) lo Storage Gateway Service API Reference.

- Per API i dettagli, vedere [ListFileShares](#) in AWS CLI Command Reference.

## list-gateways

Il seguente esempio di codice mostra come utilizzare `list-gateways`.

### AWS CLI

Per elencare i gateway per un account

Il `list-gateways` comando seguente elenca tutti i gateway definiti per un account:

```
aws storagegateway list-gateways
```

Questo comando genera un JSON blocco che contiene un elenco di gateway Amazon Resource Names (ARNs).

- Per API i dettagli, consulta [ListGateways AWS CLI Command Reference](#).

## list-volumes

Il seguente esempio di codice mostra come utilizzare `list-volumes`.

### AWS CLI

Per elencare i volumi configurati per un gateway

Il `list-volumes` comando seguente restituisce un elenco di volumi configurati per il gateway specificato. Per specificare quale gateway descrivere, usa l'Amazon Resource Name (ARN) del gateway nel comando.

Questo esempio specifica un gateway con l'id `sgw-12A3456B` nell'account `123456789012`:

```
aws storagegateway list-volumes --gateway-arn "arn:aws:storagegateway:us-west-2:123456789012:gateway/sgw-12A3456B"
```

Questo comando restituisce un JSON blocco contenente un elenco di volumi che include il tipo e ARN per ogni volume.

- Per API i dettagli, vedere [ListVolumes](#) in AWS CLI Command Reference.

## refresh-cache

Il seguente esempio di codice mostra come utilizzare `refresh-cache`.

### AWS CLI

Per aggiornare la cache di condivisione dei file

L'esempio seguente aggiorna la cache per la condivisione di file specificata.

```
aws storagegateway refresh-cache \  
  --file-share-arn arn:aws:storagegateway:us-east-1:111122223333:share/  
share-2FA12345
```

Output:

```
{  
  "FileShareARN": "arn:aws:storagegateway:us-east-1:111122223333:share/  
share-2FA12345",  
  "NotificationId": "4954d4b1-abcd-ef01-1234-97950a7d3483"  
}
```

Per ulteriori informazioni, vedere [ListFileShares AWS](#) lo Storage Gateway Service API Reference.

- Per API i dettagli, vedere [RefreshCache](#) in AWS CLI Command Reference.

## AWS STS esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS STS.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)



## Azioni

### **assume-role-with-saml**

Il seguente esempio di codice mostra come utilizzare `assume-role-with-saml`.

#### AWS CLI

Per ottenere credenziali a breve termine per un ruolo autenticato con SAML

Il `assume-role-with-saml` comando seguente recupera un set di credenziali a breve termine per il ruolo. IAM TestSam1 La richiesta in questo esempio viene autenticata utilizzando l'SAMLEncoder fornito dal provider di identità al momento dell'autenticazione.

```
aws sts assume-role-with-saml \
  --role-arn arn:aws:iam::123456789012:role/TestSam1 \
  --principal-arn arn:aws:iam::123456789012:saml-provider/SAML-test \
  --saml-assertion "VERYLONGENCODEDASSERTIONEXAMPLExYW1s0kF1ZG11bmN1PmJsYW5rPC9zYW1s0kF1ZG11bmN1Pjwv
+PHNhWw6TmFtZULEIEZvcm1hdD0idXJu0m9hc2lz0m5hbWVz0nRj0LNBTUw6Mi4w0m5hbWVpZC1mb3JtYXQ6dHJhbnM
+PHNhWw6U3ViamVjdENvbmZpcm1hdGlvb1BNZXRob2Q9InVyb3pvYXNpczpuYW1lc3p0YzpzTQU1M0jiuMDpjbTpiZWwv"
```

Output:

```
{
  "Issuer": "https://integ.example.com/idp/shibboleth</Issuer",
  "AssumedRoleUser": {
    "Arn": "arn:aws:sts::123456789012:assumed-role/TestSam1",
    "AssumedRoleId": "AR0456EXAMPLE789:TestSam1"
  },
  "Credentials": {
    "AccessKeyId": "ASIAV3ZUEFP6EXAMPLE",
    "SecretAccessKey": "8P+SQvWIuLnKhh8d++jpw0nNmQRBZvNEXAMPLEKEY",
    "SessionToken": "IQoJb3JpZ2luX2VjE0z//////////
wEXAMPLEtMSJHMEUCIDoKK3JH9uGQE1z0sINr5M4jk
+Na8KHDcCYRVjJCZEv0AiEA30vJGtw1EcVi0leS2vhs8VdCKFJQWPQrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburED
+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
+scqKmlzm8FDrypNC9Yjc8fP0Ln9FX9KSYvKTr4rvx3iS1lTJabIQwj2ICCR/oLxBA==",
    "Expiration": "2019-11-01T20:26:47Z"
  },
  "Audience": "https://signin.aws.amazon.com/saml",
  "SubjectType": "transient",
  "PackedPolicySize": "6",
```

```

    "NameQualifier": "SbdG0nUkh1i4+EXAMPLExL/jEvs=",
    "Subject": "SamlExample"
}

```

Per ulteriori informazioni, consulta [Richiesta di credenziali di sicurezza temporanee](#) nella Guida per l'utente AWS IAM

- Per API i dettagli, vedere [AssumeRoleWithSamlin](#) AWS CLI Command Reference.

## assume-role-with-web-identity

Il seguente esempio di codice mostra come utilizzare `assume-role-with-web-identity`.

### AWS CLI

Per ottenere credenziali a breve termine per un ruolo autenticato con Web Identity (OAuth2."0)

Il `assume-role-with-web-identity` comando seguente recupera un set di credenziali a breve termine per il ruolo. IAM app1 La richiesta viene autenticata utilizzando il token di identità Web fornito dal provider di identità Web specificato. Alla sessione vengono applicate due politiche aggiuntive per limitare ulteriormente ciò che l'utente può fare. Le credenziali restituite scadono un'ora dopo la generazione.

```

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"

```

Output:

```

{
  "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
+FvwqnKwRc0IfjRh3c/LTo6UDdyJw00vEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJ0gQs8IZZaIv2BXIa2R40lgkBN9bkUDNCJiBeb/
AX1zBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wn0E49mR1/+0tkIKG07fAE",
  "Expiration": "2020-05-19T18:06:10+00:00"
},
"Provider": "www.amazon.com"
}

```

Per ulteriori informazioni, vedere [Richiesta di credenziali di sicurezza temporanee](#) nella Guida per l'utente.AWS IAM

- Per API i dettagli, vedere [AssumeRoleWithWebIdentity](#) in AWS CLI Command Reference.

## assume-role

Il seguente esempio di codice mostra come utilizzare `assume-role`.

### AWS CLI

Come assumere un ruolo

Il `assume-role` comando seguente recupera un set di credenziali a breve termine per il IAM ruolo. `s3-access-example`

```

aws sts assume-role \
  --role-arn arn:aws:iam::123456789012:role/xaccounts3access \
  --role-session-name s3-access-example

```

Output:

```

{
  "AssumedRoleUser": {

```

```

    "AssumedRoleId": "ARO3XFRBF535PLBIFPI4:s3-access-example",
    "Arn": "arn:aws:sts::123456789012:assumed-role/xaccounts3access/s3-access-
example"
  },
  "Credentials": {
    "SecretAccessKey": "9drTJvcXLB89EXAMPLEL8923FB892xMFI",
    "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/qwjzP2iEXAMPLEbw/
m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtY1FVgAUj
+7Indz3LU0aTwk1WKIjHmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lfloeie3bDiNHncmzosRM6SFipzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8BRi2
IcrxSpnWEXAMPLEXSDFTAQAM6D19zR0tXoybnlrZIwML1Mi1Kcgo50ytwU=",
    "Expiration": "2016-03-15T00:05:07Z",
    "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
  }
}

```

L'output del comando contiene una chiave di accesso, una chiave segreta e un token di sessione che puoi utilizzare per l'autenticazione in AWS.

A scopo di AWS CLI utilizzo, è possibile impostare un profilo denominato associato a un ruolo. Quando usi il profilo, AWS CLI chiameranno `assume-role` e gestiranno le credenziali per te. Per ulteriori informazioni, consulta [Utilizzare un IAM ruolo nella Guida per l' AWS CLI](#) utente.AWS CLI

- Per API i dettagli, vedere [AssumeRole](#) in AWS CLI Command Reference.

## decode-authorization-message

Il seguente esempio di codice mostra come utilizzare `decode-authorization-message`.

### AWS CLI

Per decodificare un messaggio di autorizzazione codificato restituito in risposta a una richiesta

L'`decode-authorization-message` esempio seguente decodifica informazioni aggiuntive sullo stato di autorizzazione di una richiesta da un messaggio codificato restituito in risposta a una richiesta di Amazon Web Services.

```

aws sts decode-authorization-message \
  --encoded-message EXAMPLEWodyRNrtLQARDip-
eTA6i6DrLUhHhPQrLWB_LAb15pAKx19mPDLexYcGBreyIKQC1BGBIpbKxr3dFDkwqe07e2NMk5j_hmzAiChJN-8oy3Ewi

```

```
Ojau7BMj0TWw0tHPPhV_Zaz87yENDipr745EjQwRd5LaoL3vN8_5ZfA9UiBMKDgVh1gjqZJFUiQoubv78V1RbHNYnK44E
p0u3FZjwYStfvTb3GHs3-6rLribG09jZ0ktkfE6vqx1FzLyeDr4P2ihC1wty9tArCvvGzIAUNmARQJ2VWVPxioqgoqCz
JWP5pwe_mAyqh0NLw-r1S56YC_90onj9A80sNrHLI-
tIiNd7tgNTYzDuPQYD2FMDbNp82V9eVmYgtPp5NIeSpuf3f0HanFuBZgENxZQZ2dLH3xJGMTtYayzZrRXjiq_SfX9zeB
FaoPIb8LmmKVBLpIB0iFhU9sEHPqKHVPi6jdxXqKaZaFGvYVmVOiuQdNQKuyk0p067POFrZECLjj0tNPB0ZCcuEKEXAM
```

## Output:

```
{
  "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":true,\"matchedStatements\
\":{\
\"items\
\":[\
\"statementId\
\":\
\"VisualEditor0\
\",
\"effect\
\":\
\"DENY\
\",
\"principals\
\":{\
\"items\
\":[\
\"value\
\":\
\"AROA123456789EXAMPLE\
\"]}],
\"principalGroups\
\":{\
\"items\
\":[\
\"]}],
\"actions\
\":{\
\"items\
\":[\
\"value\
\":\
\"ec2:RunInstances\
\"]}],
\"resources\
\":{\
\"items\
\":[\
\"value\
\":\
\"*\
\"]}],
\"conditions\
\":{\
\"items\
\":[\
\"]}],
\"failures\
\":{\
\"items\
\":[\
\"]}],
\"context\
\":{\
\"principal\
\":{\
\"id\
\":\
\"AROA123456789EXAMPLE:Ana\
\",
\"arn\
\":\
\"arn:aws:sts::111122223333:assumed-role/Developer/Ana\
\"},
\"action\
\":\
\"RunInstances\
\",
\"resource\
\":\
\"arn:aws:ec2:us-east-1:111122223333:instance/*\
\",
\"conditions\
\":{\
\"items\
\":[\
\"key\
\":\
\"ec2:MetadataHttpResponseHopLimit\
\",
\"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: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\
\":\
\"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/*\
\"]}}}}}"
}
```

Per ulteriori informazioni, consulta [Logica di valutazione delle politiche](#) nella Guida per l'AWS IAMutente.

- Per API i dettagli, vedere [DecodeAuthorizationMessage](#) in AWS CLI Command Reference.

## get-caller-identity

Il seguente esempio di codice mostra come utilizzare `get-caller-identity`.

### AWS CLI

Per ottenere dettagli sull'IAM identità corrente

Il `get-caller-identity` comando seguente visualizza informazioni sull'IAM identità utilizzata per autenticare la richiesta. Il chiamante è un IAM utente.

```
aws sts get-caller-identity
```

Output:

```
{
  "UserId": "AIDASAMPLEUSERID",
  "Account": "123456789012",
  "Arn": "arn:aws:iam::123456789012:user/DevAdmin"
}
```

- Per API i dettagli, vedere [GetCallerIdentity](#) in AWS CLI Command Reference.

## get-federation-token

Il seguente esempio di codice mostra come utilizzare `get-federation-token`.

### AWS CLI

Per restituire un set di credenziali di sicurezza temporanee utilizzando le credenziali della chiave di accesso IAM utente

L'`get-federation-token` esempio seguente restituisce un set di credenziali di sicurezza temporanee (costituite da un ID di chiave di accesso, una chiave di accesso segreta e un token di sicurezza) per un utente. È necessario chiamare l'`GetFederationToken` operazione utilizzando le credenziali di sicurezza a lungo termine di un IAM utente.

```
aws sts get-federation-token \
  --name Bob \
  --policy file://myfile.json \
  --policy-arns arn=arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess \
```

```
--duration-seconds 900
```

Contenuto di 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": "*"
    }
  ]
}
```

Output:

```
{
  "Credentials": {
    "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
    "SessionToken": "EXAMPLEpZ21uX2VjEGoaCXVzLXd1c3QtMiJIMEYCIQC/
W9pL5ArQyDD5JwFL3/h5+WGopQ24GEXweNctwhi9sgIhAMkg
+MZE35iWM8s4r5Lr25f9rSTVPFH98G42QunWMTfKq0DCOP//////////"
```

```
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
}
```

Per ulteriori informazioni, vedere [Richiesta di credenziali di sicurezza temporanee nella Guida per l'AWS IAM](#) utente.

- Per API i dettagli, vedere [GetFederationToken](#) in AWS CLI Command Reference.

## get-session-token

Il seguente esempio di codice mostra come utilizzare `get-session-token`.

### AWS CLI

Per ottenere un set di credenziali a breve termine per un'identità IAM

Il `get-session-token` comando seguente recupera un set di credenziali a breve termine per l'IAM identità che effettua la chiamata. Le credenziali risultanti possono essere utilizzate per le richieste in cui l'autenticazione a più fattori (MFA) è richiesta dalla policy. Le credenziali scadono 15 minuti dopo la loro generazione.

```
aws sts get-session-token \
  --duration-seconds 900 \
  --serial-number "YourMFADeviceSerialNumber" \
  --token-code 123456
```

Output:

```
{
  "Credentials": {
    "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYzEXAMPLEKEY",
    "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4B1CFFxWNE1OPTgk5TthT
+FvqwqKwRc0IfrRh3c/LTo6UDdyJw00vEVPvLXCrrrUtdnniCEXAMPLE/"
  }
}
```



```
IvU1dYUg2RVAJBanLiHb4IgrmpRV3zrkuWJ0gQs8IZZaIv2BXIa2R40l9kBN9bkUDNCJiBeb/  
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wn0E49mRl/+0tkIKG07fAE",  
    "Expiration": "2020-05-19T18:06:10+00:00"  
  }  
}
```

Per ulteriori informazioni, consulta [Richiesta di credenziali di sicurezza temporanee](#) nella Guida per l'utente AWS IAM

- Per API i dettagli, vedere [GetSessionToken](#) in AWS CLI Command Reference.

## AWS Support esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS Support.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **add-attachments-to-set**

Il seguente esempio di codice mostra come utilizzare `add-attachments-to-set`.

#### AWS CLI

Per aggiungere un allegato a un set

L'esempio seguente aggiunge un'immagine a un set che potete quindi specificare per una richiesta di supporto nel vostro AWS account.

```
aws support add-attachments-to-set \
```

```
--attachment-set-id "as-2f5a6faa2a4a1e600-mu-nk5xQ1Br70-  
G1cUos5LZkd38K0AHZa9BMDVzNEXAMPLE" \  
--attachments fileName=troubleshoot-screenshot.png,data=base64-encoded-string
```

Output:

```
{  
  "attachmentSetId": "as-2f5a6faa2a4a1e600-mu-nk5xQ1Br70-  
G1cUos5LZkd38K0AHZa9BMDVzNEXAMPLE",  
  "expiryTime": "2020-05-14T17:04:40.790+0000"  
}
```

Per ulteriori informazioni, consulta la sezione [Gestione dei casi](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [AddAttachmentsToSet](#) in AWS CLI Command Reference.

## add-communication-to-case

Il seguente esempio di codice mostra come utilizzare `add-communication-to-case`.

AWS CLI

Per aggiungere una comunicazione a un caso

L'`add-communication-to-case` esempio seguente aggiunge comunicazioni a un caso di supporto nel tuo AWS account.

```
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"
```

Output:

```
{  
  "result": true  
}
```

Per ulteriori informazioni, consulta la sezione [Gestione dei casi](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [AddCommunicationToCase](#) in AWS CLI Command Reference.

## create-case

Il seguente esempio di codice mostra come utilizzare `create-case`.

### AWS CLI

Per creare un caso

L'`create-case` esempio seguente crea una richiesta di supporto per il tuo AWS account.

```
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"
```

Output:

```
{  
  "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47"  
}
```

Per ulteriori informazioni, consulta la sezione [Gestione dei casi](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [CreateCase](#) in AWS CLI Command Reference.

## describe-attachment

Il seguente esempio di codice mostra come utilizzare `describe-attachment`.

### AWS CLI

Per descrivere un allegato

L'`describe-attachment` esempio seguente restituisce informazioni sull'allegato con l'ID specificato.

```
aws support describe-attachment \  
  --attachment-id "attachment-KBnjRNrePd9D6Jx0-Mm00xZuDEaL2JAj_0-  
gJv9qqDooTipsz3V1Nb19rCfkZneeQeDPgp8X1iVJyHH7UuhZDdNeqGoduZsPrAhyMakq1c60-  
iJjL5HqyYGiT1FG8EXAMPLE"
```

Output:

```
{  
  "attachment": {  
    "fileName": "troubleshoot-screenshot.png",  
    "data": "base64-blob"  
  }  
}
```

Per ulteriori informazioni, consulta la sezione [Gestione dei casi](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [DescribeAttachment](#) in AWS CLI Command Reference.

## describe-cases

Il seguente esempio di codice mostra come utilizzare `describe-cases`.

### AWS CLI

Descrivere un caso

L'`describe-cases` esempio seguente restituisce informazioni sulla richiesta di assistenza specificata nel tuo AWS account.

```
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
```

Output:

```
{  
  "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"
  }
]
}
```

Per ulteriori informazioni, consulta la sezione [Gestione dei casi](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [DescribeCases](#) in AWS CLI Command Reference.

## describe-communications

Il seguente esempio di codice mostra come utilizzare `describe-communications`.

### AWS CLI

Descrivere la comunicazione più recente relativa a un caso

L'`describe-communications` esempio seguente restituisce la comunicazione più recente per il caso di assistenza specificato nel tuo AWS account.

```
aws support describe-communications \
  --case-id "case-12345678910-2013-c4c1d2bf33c5cf47" \
  --after-time "2020-03-23T21:31:47.774Z" \
  --max-item 1
```

Output:

```
{
  "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": "eyJuZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQEXAMPLE=="
  }

```

Per ulteriori informazioni, consulta la sezione [Gestione dei casi](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [DescribeCommunications](#) in AWS CLI Command Reference.

## describe-services

Il seguente esempio di codice mostra come utilizzare `describe-services`.

### AWS CLI

Per elencare AWS i servizi e le categorie di servizi

L'esempio seguente elenca le categorie di servizi disponibili per la richiesta di informazioni generali.

```

aws support describe-services \
  --service-code-list "general-info"

```

Output:

```

{
  "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"
      }
    ]
  }
]
```

Per ulteriori informazioni, consulta la sezione [Gestione dei casi](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [DescribeServices](#) in AWS CLI Command Reference.

## describe-severity-levels

Il seguente esempio di codice mostra come utilizzare `describe-severity-levels`.

### AWS CLI

Per elencare i livelli di gravità disponibili

L'esempio seguente elenca i livelli di gravità disponibili per un caso di supporto.

```
aws support describe-severity-levels
```

**Output:**

```
{
  "severityLevels": [
    {
      "code": "low",
      "name": "Low"
    },
    {
      "code": "normal",
      "name": "Normal"
    },
    {
      "code": "high",
      "name": "High"
    },
    {
      "code": "urgent",
      "name": "Urgent"
    },
    {
      "code": "critical",
      "name": "Critical"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Scelta della gravità](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [DescribeSeverityLevels](#) in AWS CLI Command Reference.

**describe-trusted-advisor-check-refresh-statuses**

Il seguente esempio di codice mostra come utilizzare `describe-trusted-advisor-check-refresh-statuses`.

**AWS CLI**

Per elencare gli stati di aggiornamento dei controlli di AWS Trusted Advisor

L'esempio seguente elenca gli stati di aggiornamento per due controlli Trusted Advisor: Amazon S3 Bucket Permissions e Use IAM



```
aws support describe-trusted-advisor-check-refresh-statuses \  
--check-id "Pfx0RwqBli" "zXCkfM1nI3"
```

Output:

```
{  
  "statuses": [  
    {  
      "checkId": "Pfx0RwqBli",  
      "status": "none",  
      "millisUntilNextRefreshable": 0  
    },  
    {  
      "checkId": "zXCkfM1nI3",  
      "status": "none",  
      "millisUntilNextRefreshable": 0  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [AWS Trusted Advisor](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [DescribeTrustedAdvisorCheckRefreshStatuses](#) in AWS CLI Command Reference.

## describe-trusted-advisor-check-result

Il seguente esempio di codice mostra come utilizzare `describe-trusted-advisor-check-result`.

AWS CLI

Per elencare i risultati di un controllo di AWS Trusted Advisor

L'`describe-trusted-advisor-check-result` esempio seguente elenca i risultati del controllo IAM Use.

```
aws support describe-trusted-advisor-check-result \  
--check-id "zXCkfM1nI3"
```

Output:

```
{
  "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
      }
    ]
  }
}
```

Per ulteriori informazioni, consulta [AWS Trusted Advisor](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [DescribeTrustedAdvisorCheckResult](#) in AWS CLI Command Reference.

## describe-trusted-advisor-check-summaries

Il seguente esempio di codice mostra come utilizzare `describe-trusted-advisor-check-summaries`.

### AWS CLI

Per elencare i riepiloghi dei controlli di AWS Trusted Advisor

L'esempio seguente elenca i risultati di due controlli Trusted Advisor: Amazon S3 Bucket Permissions e Use. IAM

```
aws support describe-trusted-advisor-check-summaries \  
--check-ids "Pfx0RwqBli" "zXCkfM1nI3"
```

Output:

```
{  
  "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  
        }  
      }  
    }  
  ]  
}
```

```
]
}
```

Per ulteriori informazioni, consulta [AWS Trusted Advisor](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [DescribeTrustedAdvisorCheckSummaries](#) in AWS CLI Command Reference.

## describe-trusted-advisor-checks

Il seguente esempio di codice mostra come utilizzare `describe-trusted-advisor-checks`.

### AWS CLI

Per elencare i controlli AWS Trusted Advisor disponibili

L'esempio seguente elenca gli assegni Trusted Advisor disponibili nel tuo AWS account. Queste informazioni includono il nome, l'ID, la descrizione, la categoria e i metadati dell'assegno. Nota che l'output è abbreviato per motivi di leggibilità.

```
aws support describe-trusted-advisor-checks \
  --language "en"
```

Output:

```
{
  "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": []
    }
  ]
}
```

```
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [AWS Trusted Advisor](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [DescribeTrustedAdvisorChecks](#) in AWS CLI Command Reference.

## refresh-trusted-advisor-check

Il seguente esempio di codice mostra come utilizzare `refresh-trusted-advisor-check`.

### AWS CLI

Per aggiornare un controllo di AWS Trusted Advisor

L'esempio seguente aggiorna il check di Amazon S3 Bucket Permissions Trusted Advisor nel tuo account. AWS

```
aws support refresh-trusted-advisor-check \  
  --check-id "Pfx0RwqBli"
```

Output:

```
{  
  "status": {  
    "checkId": "Pfx0RwqBli",  
    "status": "enqueued",  
    "millisUntilNextRefreshable": 3599992  
  }  
}
```

Per ulteriori informazioni, consulta [AWS Trusted Advisor](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [RefreshTrustedAdvisorCheck](#) in AWS CLI Command Reference.

## resolve-case

Il seguente esempio di codice mostra come utilizzare `resolve-case`.

### AWS CLI

Per risolvere un caso di assistenza

L'`resolve-case` esempio seguente risolve un caso di assistenza nel tuo AWS account.

```
aws support resolve-case \  
  --case-id "case-12345678910-2013-c4c1d2bf33c5cf47"
```

Output:

```
{  
  "finalCaseStatus": "resolved",  
  "initialCaseStatus": "work-in-progress"  
}
```

Per ulteriori informazioni, consulta la sezione [Gestione dei casi](#) nella AWS Support User Guide.

- Per API i dettagli, vedere [ResolveCase](#) in AWS CLI Command Reference.

## SWF Esempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con AmazonSWF.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **count-closed-workflow-executions**

Il seguente esempio di codice mostra come utilizzare `count-closed-workflow-executions`.

AWS CLI

Conteggio delle esecuzioni di workflow chiuse

È possibile utilizzarlo `swf count-closed-workflow-executions` per recuperare il numero di esecuzioni di workflow chiuse per un determinato dominio. È possibile specificare filtri per contare classi specifiche di esecuzioni.

Gli `--start-time-filter` argomenti `--domain` e either `--close-time-filter` or sono obbligatori. Tutti gli altri argomenti sono facoltativi.

```
aws swf count-closed-workflow-executions \
  --domain DataFrobtzz \
  --close-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :
  1370044800 }"
```

Output:

```
{
  "count": 2,
  "truncated": false
}
```

Se «troncato» è `true`, allora «count» rappresenta il numero massimo che può essere restituito da Amazon. SWF Eventuali ulteriori risultati vengono troncati.

Per ridurre il numero di risultati restituiti, puoi:

modificare i `--start-time-filter` valori `--close-time-filter` o per restringere l'intervallo di tempo in cui viene effettuata la ricerca. Ciascuno di questi si esclude a vicenda: puoi specificarne solo uno in una richiesta. Utilizza gli argomenti `--close-status-filter`, `--execution-filter`, `--tag-filter` o `--type-filter` per filtrare ulteriormente i risultati. Tuttavia, anche questi argomenti si escludono a vicenda.

Vedi anche [CountClosedWorkflowExecutions](#) nella pagina di API riferimento del servizio Amazon Simple Workflow

- Per API i dettagli, consulta [CountClosedWorkflowExecutions AWS CLI Command Reference](#).

## **count-open-workflow-executions**

Il seguente esempio di codice mostra come utilizzare `count-open-workflow-executions`.

AWS CLI

Conteggio delle esecuzioni di flussi di lavoro aperti

È possibile utilizzarlo `swf count-open-workflow-executions` per recuperare il numero di esecuzioni di workflow aperte per un determinato dominio. È possibile specificare filtri per contare classi specifiche di esecuzioni.

Gli `--start-time-filter` argomenti `--domain` and sono obbligatori. Tutti gli altri argomenti sono facoltativi.

```
aws swf count-open-workflow-executions \
  --domain DataFrobtzz \
  --start-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :
1370044800 }"
```

Output:

```
{
  "count": 4,
  "truncated": false
}
```

Se «troncato» è `true`, allora «count» rappresenta il numero massimo che può essere restituito da Amazon. SWF Eventuali ulteriori risultati vengono troncati.

Per ridurre il numero di risultati restituiti, puoi:

modificare i `--start-time-filter` valori per restringere l'intervallo di tempo in cui viene effettuata la ricerca. Utilizzare gli `--type-filter` argomenti `--close-status-filter`, `--execution-filter`, `--tag-filter` o per filtrare ulteriormente i risultati. Ciascuno di questi si esclude a vicenda: è possibile specificare solo uno di questi in una richiesta.

Per ulteriori informazioni, consulta [CountOpenWorkflowExecutions Amazon Simple Workflow Service API Reference](#)

- Per API i dettagli, consulta [CountOpenWorkflowExecutions AWS CLI Command Reference](#).

## **deprecate-domain**

Il seguente esempio di codice mostra come utilizzare `deprecate-domain`.

### AWS CLI

Deprecare un dominio



Per impostare come obsoleto un dominio (che resta comunque visibile, ma non può essere utilizzato per creare nuove esecuzioni di flussi di lavoro o registrare tipi), utilizza `swf deprecate-domain`. Esiste un solo parametro obbligatorio, `--name`, che assume il nome del dominio da dichiarare obsoleto.

```
aws swf deprecate-domain \  
  --name MyNeatNewDomain ""
```

Come per `register-domain`, non viene restituito alcun output. Se utilizzi `list-domains` per visualizzare i domini registrati, tuttavia, vedrai che il dominio è diventato obsoleto e non appare più nei dati restituiti.

```
aws swf list-domains \  
  --registration-status REGISTERED \  
  {  
    "domainInfos": [  
      {  
        "status": "REGISTERED",  
        "name": "DataFrobotz"  
      },  
      {  
        "status": "REGISTERED",  
        "name": "erontest"  
      }  
    ]  
  }
```

Se usi `--registration-status DEPRECATED` with `list-domains`, vedrai il tuo dominio obsoleto.

```
aws swf list-domains \  
  --registration-status DEPRECATED \  
  {  
    "domainInfos": [  
      {  
        "status": "DEPRECATED",  
        "name": "MyNeatNewDomain"  
      }  
    ]  
  }
```

Puoi comunque utilizzarlo `describe-domain` per ottenere informazioni su un dominio obsoleto.

```
aws swf describe-domain \  
  --name MyNeatNewDomain  
  {  
    "domainInfo": {  
      "status": "DEPRECATED",  
      "name": "MyNeatNewDomain"  
    },  
    "configuration": {  
      "workflowExecutionRetentionPeriodInDays": "0"  
    }  
  }
```

Vedi anche [DeprecateDomain](#) nella pagina di API riferimento del servizio Amazon Simple Workflow

- Per API i dettagli, consulta [DeprecateDomain AWS CLI Command Reference](#).

## describe-domain

Il seguente esempio di codice mostra come utilizzare `describe-domain`.

### AWS CLI

Ottenere informazioni su un dominio

Per ottenere informazioni dettagliate su un particolare dominio, usa il `swf describe-domain` comando. Esiste un solo parametro obbligatorio: `--name`, che assume il nome del dominio su cui si desidera ottenere informazioni.

```
aws swf describe-domain \  
  --name DataFrobotz  
  {  
    "domainInfo": {  
      "status": "REGISTERED",  
      "name": "DataFrobotz"  
    },  
    "configuration": {  
      "workflowExecutionRetentionPeriodInDays": "1"  
    }  
  }
```

Puoi anche usarlo `describe-domain` per ottenere informazioni sui domini obsoleti.

```
aws swf describe-domain \  
  --name MyNeatNewDomain \  
  {  
    "domainInfo": {  
      "status": "DEPRECATED",  
      "name": "MyNeatNewDomain"  
    },  
    "configuration": {  
      "workflowExecutionRetentionPeriodInDays": "0"  
    }  
  }  
}
```

Vedi anche [DescribeDomain](#) nella pagina di API riferimento del servizio Amazon Simple Workflow

- Per API i dettagli, consulta [DescribeDomain AWS CLI Command Reference](#).

## list-activity-types

Il seguente esempio di codice mostra come utilizzare `list-activity-types`.

### AWS CLI

Elencare i tipi di attività

Per ottenere un elenco dei tipi di attività per un dominio, usa `swf list-activity-types`. Gli `--registration-status` argomenti `--domain` e sono obbligatori.

```
aws swf list-activity-types \  
  --domain DataFrobtzz \  
  --registration-status REGISTERED
```

Output:

```
{  
  "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"
  }
]
}
```

È possibile utilizzare l'`--name` argomento per selezionare solo i tipi di attività con un nome particolare:

```
aws swf list-activity-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED \
  --name "send-subscription-success"
```

Output:

```
{
  "typeInfos": [
    {
      "status": "REGISTERED",
      "creationDate": 1371454150.909,
      "activityType": {
        "version": "1",
        "name": "send-subscription-success"
      },
      "description": "subscribe send-subscription-success activity"
    }
  ]
}
```

Per recuperare i risultati nelle pagine, puoi impostare l'`--maximum-page-size` argomento. Se vengono restituiti più risultati di quelli contenuti in una pagina di risultati, verrà restituito un `nextPageToken` "" nel set di risultati:

```
aws swf list-activity-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED \
  --maximum-page-size 2
```

Output:

```
{
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAA1Gp1Be1Jq
+PmHvAnDxJYbup8+0R4LVtbXLD17QNY7C30pHo9Sx06D/GuFz10yC73umBQ1t0PJ/gC/
aYpzDMqUIWIA1T9W0s2DryyZX40C/6Lhk9/
o5kdsuWMSBkHhgaZjgwp3WJINIFJFdaSMxY2vYAX7AtRtpcqJuBDDRE9RaRqDGYqIYUM1tarkiqpSY1ZVveBasBv1vyU
WGAaqehiDz7/JzLT/wWNNUM0d+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"
    }
  ]
}

```

È possibile passare il nextPageToken valore alla chiamata successiva `list-activity-types` all'`--next-page-token` argomento, recuperando la pagina successiva di risultati:

```

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 qpSY1ZVveBasBvLvyl
WGAAqehiDz7/JzLT/wWNNUM0d+Nhe"

```

Output:

```

{
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAAAAw+7LZ4GRZPzTqBHsp2wBxWB8m1sgLCc1gCuq3J+h/
m3+v0fFqtkcjLwV5cc40jNAzTCuq/
XcylPumGwkjbajtqpZpbq0cVNfjFxFxGoi0LB201bv0krbUISBvlpFPmSwpDSZJxsxg5UxCcweteS1Fn1PNSZ/
MoinBZo80TkjMuzcsTuK0zH9wCaR8ITcALJ3SaqHU3pyIRS5hPmFA30LIc8zaAepjlaujo6hntNSCruB4"
  "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"
  }
]
}

```

Se ci sono ancora altri risultati da restituire, verrà restituito `nextPageToken` "" con i risultati. Quando non ci sono più pagine di risultati da restituire, `nextPageToken` non verrà restituito nel set di risultati.

È possibile utilizzare l' `--reverse-order` argomento per invertire l'ordine dei risultati restituiti. Ciò influisce anche sui risultati paginati.

```

aws swf list-activity-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED \
  --maximum-page-size 2 \
  --reverse-order

```

Output:

```

{
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAwXcpu5ePSyQkrC
+8WMbmSrenuZC2ZkIXQYBPB/b9xIOVkj+bMEFhGj0KmmJ4rF7iddhjf7UMYCsfGkEn7mk
+yMCgVc1JxDWmB0EH46bhcmclmYNQihMDmUwocpr7To6/R7CLu0St1gkFayx0idJXErQW0zdNfQaIWAnF/
cwioBbXlkz1fQzmDeU3M5oYGMPQIrUqkPq7pMEW0q0lK5eDN97NzFYdZZ/r1cLDWPZhUjY",
  "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"
    }
]
}

```

Vedi anche [ListActivityTypes](#) nella pagina di API riferimento del servizio Amazon Simple Workflow

- Per API i dettagli, consulta [ListActivityTypes AWS CLI Command Reference](#).

## list-domains

Il seguente esempio di codice mostra come utilizzare `list-domains`.

### AWS CLI

Esempio 1: Per elencare i domini registrati

Il seguente esempio di `list-domains` comando elenca i REGISTERED SWF domini che hai registrato per il tuo account.

```

aws swf list-domains \
  --registration-status REGISTERED

```

Output:

```

{
  "domainInfos": [
    {

```



```
    "status": "REGISTERED",
    "name": "DataFrobotz"
  },
  {
    "status": "REGISTERED",
    "name": "erontest"
  }
]
```

Per ulteriori informazioni, consulta [ListDomains](#) Amazon Simple Workflow Service API Reference

Esempio 2: per elencare i tuoi domini obsoleti

Il seguente esempio di `list-domains` comando elenca i DEPRECATED SWF domini che hai registrato per il tuo account. I domini obsoleti sono domini che non possono registrare nuovi flussi di lavoro o attività, ma che possono comunque essere interrogati.

```
aws swf list-domains \
  --registration-status DEPRECATED
```

Output:

```
{
  "domainInfos": [
    {
      "status": "DEPRECATED",
      "name": "MyNeatNewDomain"
    }
  ]
}
```

Per ulteriori informazioni, consulta [ListDomains](#) Amazon Simple Workflow Service API Reference

Esempio 3: per elencare la prima pagina dei domini registrati

Il seguente esempio di `list-domains` comando elenca i REGISTERED SWF domini della prima pagina che hai registrato per il tuo account utilizzando l'`--maximum-page-size` opzione.

```
aws swf list-domains \
  --registration-status REGISTERED \
  --maximum-page-size 1
```

## Output:

```
{
  "domainInfos": [
    {
      "status": "REGISTERED",
      "name": "DataFrobotz"
    }
  ],
  "nextPageToken": "AAAAKgAAAAEAAAAAAAAAA2QJKNtidVgd49TTeNwYcpD
+QKT2ynuEbibcQWe2QKrslMGe63gpS0MgZGpcpoKttL40CXRFn98Xif557it
+wSZUsvUDtImjDLvguyuyyFdzIZtvIxIKE0Pm3k2r40jAGaFsG0uVbrKljv1a7wdU7FYH301kNCP8b7PBj9SBkUyGoiAg"
}
```

Per ulteriori informazioni, consulta [ListDomains](#) Amazon Simple Workflow Service API Reference

Esempio 4: per elencare la pagina singola specificata dei domini registrati

Il seguente esempio di `list-domains` comando elenca i REGISTERED SWF domini della prima pagina che hai registrato per il tuo account utilizzando l'`--maximum-page-size` opzione.

Quando effettui nuovamente la chiamata, questa volta inserendo il valore di `nextPageToken` nell'`--next-page-token` argomento, otterrai un'altra pagina di risultati.

```
aws swf list-domains \
  --registration-status REGISTERED \
  --maximum-page-size 1 \
  --next-page-token "AAAAKgAAAAEAAAAAAAAAA2QJKNtidVgd49TTeNwYcpD
+QKT2ynuEbibcQWe2QKrslMGe63gpS0MgZGpcpoKttL40CXRFn98Xif557it
+wSZUsvUDtImjDLvguyuyyFdzIZtvIxIKE0Pm3k2r40jAGaFsG0uVbrKljv1a7wdU7FYH301kNCP8b7PBj9SBkUyGoiAg"
```

## Output:

```
{
  "domainInfos": [
    {
      "status": "REGISTERED",
      "name": "erontest"
    }
  ]
}
```

Quando le pagine dei risultati da recuperare saranno terminate, `nextPageToken` non verrà restituito nei risultati.

Per ulteriori informazioni, consulta [ListDomains](#) Amazon Simple Workflow Service API Reference

- Per API i dettagli, consulta [ListDomains AWS CLI](#) Command Reference.

## list-workflow-types

Il seguente esempio di codice mostra come utilizzare `list-workflow-types`.

### AWS CLI

Elenco dei tipi di flusso di lavoro

Per ottenere un elenco dei tipi di flusso di lavoro per un dominio, usa `swf list-workflow-types`. Gli `--registration-status` argomenti `--domain` e sono obbligatori. Ecco un semplice esempio.

```
aws swf list-workflow-types \
  --domain DataFrobtzz \
  --registration-status REGISTERED
```

Output:

```
{
  "typeInfos": [
    {
      "status": "REGISTERED",
      "creationDate": 1371454149.598,
      "description": "DataFrobtzz subscribe workflow",
      "workflowType": {
        "version": "v3",
        "name": "subscribe"
      }
    }
  ]
}
```

Ad esempio `list-activity-types`, è possibile utilizzare l' `--name` argomento per selezionare solo i tipi di flusso di lavoro con un nome particolare e utilizzare l' `--maximum-page-`

sizeargomento in coordinamento con `--next-page-token` i risultati della pagina. Per invertire l'ordine in cui vengono restituiti i risultati, utilizzare `--reverse-order`.

Vedi anche [ListWorkflowTypes](#) nella pagina di API riferimento del servizio Amazon Simple Workflow

- Per API i dettagli, consulta [ListWorkflowTypes AWS CLI Command Reference](#).

## register-domain

Il seguente esempio di codice mostra come utilizzare `register-domain`.

### AWS CLI

Registrazione di un dominio

Puoi usare il AWS CLI per registrare nuovi domini. Utilizza il comando `swf register-domain`. Esistono due parametri obbligatori `--name`, che richiedono il nome di dominio e `--workflow-execution-retention-period-in-days` un numero intero per specificare il numero di giorni di conservazione dei dati di esecuzione del flusso di lavoro su questo dominio, fino a un periodo massimo di 90 giorni (per ulteriori informazioni, consulta SWF FAQ < [https://aws.amazon.com/swf/faqs/#retain\\_limit](https://aws.amazon.com/swf/faqs/#retain_limit) >). I dati di esecuzione del flusso di lavoro non verranno conservati dopo il numero di giorni specificato.

```
aws swf register-domain \  
  --name MyNeatNewDomain \  
  --workflow-execution-retention-period-in-days 0  
  ""
```

Quando si registra un dominio, non viene restituito alcun valore (""), ma è possibile utilizzare `swf list-domains` o `swf describe-domain` per visualizzare il nuovo dominio.

```
aws swf list-domains \  
  --registration-status REGISTERED  
  {  
    "domainInfos": [  
      {  
        "status": "REGISTERED",  
        "name": "DataFrobotz"  
      },  
      {
```

```

        "status": "REGISTERED",
        "name": "MyNeatNewDomain"
    },
    {
        "status": "REGISTERED",
        "name": "erontest"
    }
]
}

```

Con `swf describe-domain`:

```

aws swf describe-domain --
name MyNeatNewDomain
{
  "domainInfo": {
    "status": "REGISTERED",
    "name": "MyNeatNewDomain"
  },
  "configuration": {
    "workflowExecutionRetentionPeriodInDays": "0"
  }
}

```

Vedi anche [RegisterDomain](#) nella pagina di API riferimento del servizio Amazon Simple Workflow

- Per API i dettagli, consulta [RegisterDomain AWS CLI Command Reference](#).

## register-workflow-type

Il seguente esempio di codice mostra come utilizzare `register-workflow-type`.

### AWS CLI

Registrazione di un tipo di workflow

Per registrare un tipo di workflow con AWS CLI, utilizzare il `swf register-workflow-type` comando.

```

aws swf register-workflow-type \
  --domain DataFrobtzz \
  --name "MySimpleWorkflow" \

```

```
--workflow-version "v1"
```

In caso di successo, il comando non produce alcun output.

In caso di errore (ad esempio, se si tenta di registrare due volte lo stesso tipo di flusso di lavoro o si specifica un dominio che non esiste) si otterrà una risposta in JSON

```
{
  "message": "WorkflowType=[name=MySimpleWorkflow, version=v1]",
  "__type": "com.amazonaws.swf.base.model#TypeAlreadyExistsFault"
}
```

`--domain`, `--name` e sono obbligatori. `--workflow-version` È inoltre possibile impostare la descrizione del flusso di lavoro, i timeout e la politica del flusso di lavoro secondario.

Per ulteriori informazioni, consulta [RegisterWorkflowType](#) Amazon Simple Workflow Service API Reference

- Per API i dettagli, consulta [RegisterWorkflowType AWS CLI](#) Command Reference.

## Esempi di Systems Manager che utilizzano AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Systems Manager.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **add-tags-to-resource**

Il seguente esempio di codice mostra come utilizzare `add-tags-to-resource`.

## AWS CLI

Esempio 1: Per aggiungere tag a una finestra di manutenzione

L'add-tags-to-resourceesempio seguente aggiunge un tag alla finestra di manutenzione specificata.

```
aws ssm add-tags-to-resource \  
  --resource-type "MaintenanceWindow" \  
  --resource-id "mw-03eb9db428EXAMPLE" \  
  --tags "Key=Stack,Value=Production"
```

Questo comando non produce alcun output.

Esempio 2: per aggiungere tag a un parametro

L'add-tags-to-resourceesempio seguente aggiunge due tag al parametro specificato.

```
aws ssm add-tags-to-resource \  
  --resource-type "Parameter" \  
  --resource-id "My-Parameter" \  
  --tags '[{"Key": "Region", "Value": "East"}, {"Key": "Environment",  
  "Value": "Production"}]'
```

Questo comando non produce alcun output.

Esempio 3: aggiungere tag a un SSM documento

L'add-tags-to-resourceesempio seguente aggiunge un tag al documento specificato.

```
aws ssm add-tags-to-resource \  
  --resource-type "Document" \  
  --resource-id "My-Document" \  
  --tags "Key=Quarter,Value=Q322"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Tagging Systems Manager resources](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [AddTagsToResource](#) in AWS CLI Command Reference.

## associate-ops-item-related-item

Il seguente esempio di codice mostra come utilizzare `associate-ops-item-related-item`.

### AWS CLI

Per associare un elemento correlato

L'esempio seguente associa un elemento correlato a un `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"
```

Output:

```
{
  "AssociationId": "61d7178d-a30d-4bc5-9b4e-a9e74EXAMPLE"
}
```

Per ulteriori informazioni, vedere [Utilizzo degli incidenti di Incident Manager OpsCenter nella Guida per l'utente di AWS Systems Manager](#).

- Per API i dettagli, vedere [AssociateOpsItemRelatedItem](#) in AWS CLI Command Reference.

## cancel-command

Il seguente esempio di codice mostra come utilizzare `cancel-command`.

### AWS CLI

Esempio 1: annullare un comando per tutte le istanze

L'esempio seguente tenta di annullare il comando specificato che è già in esecuzione per tutte le istanze.

```
aws ssm cancel-command \
  --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
```



Questo comando non produce alcun output.

Esempio 2: annullare un comando per istanze specifiche

L'cancel-commandesempio seguente tenta di annullare un comando solo per l'istanza specificata.

```
aws ssm cancel-command \  
  --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE" \  
  --instance-ids "i-02573cafcfEXAMPLE"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Tagging Systems Manager Parameters](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [CancelCommand](#) in AWS CLI Command Reference.

## cancel-maintenance-window-execution

Il seguente esempio di codice mostra come utilizzarecancel-maintenance-window-execution.

### AWS CLI

Per annullare l'esecuzione di una finestra di manutenzione

Questo cancel-maintenance-window-execution esempio interrompe l'esecuzione della finestra di manutenzione specificata che è già in corso.

```
aws ssm cancel-maintenance-window-execution \  
  --window-execution-id j218d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE
```

Output:

```
{  
  "WindowExecutionId": "j218d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE"  
}
```

Per ulteriori informazioni, vedere [Systems Manager Maintenance Windows Tutorials \(AWS CLI\)](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [CancelMaintenanceWindowExecution](#) in AWS CLI Command Reference.

## create-activation

Il seguente esempio di codice mostra come utilizzare `create-activation`.

### AWS CLI

Per creare un'attivazione di un'istanza gestita

L'`create-activation` esempio seguente crea un'attivazione di un'istanza gestita.

```
aws ssm create-activation \
  --default-instance-name "HybridWebServers" \
  --iam-role "HybridWebServersRole" \
  --registration-limit 5
```

Output:

```
{
  "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
  "ActivationCode": "dRmgnYaFv567vEXAMPLE"
}
```

Per ulteriori informazioni, vedere [Passaggio 4: Creare un'attivazione di istanze gestite per un ambiente ibrido](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [CreateActivation](#) in AWS CLI Command Reference.

## create-association-batch

Il seguente esempio di codice mostra come utilizzare `create-association-batch`.

### AWS CLI

Per creare più associazioni

Questo esempio associa un documento di configurazione a più istanze. L'output restituisce un elenco delle operazioni riuscite e non riuscite, se applicabile.

Comando:

```
aws ssm create-association-batch --entries "Name=AWS-UpdateSSMAgent,InstanceId=i-1234567890abcdef0" "Name=AWS-UpdateSSMAgent,InstanceId=i-9876543210abcdef0"
```

Output:

```
{
  "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": []
}

```

- Per API i dettagli, vedere [CreateAssociationBatch](#) in AWS CLI Command Reference.

## create-association

Il seguente esempio di codice mostra come utilizzare `create-association`.

### AWS CLI

Esempio 1: associare un documento utilizzando un'istanza IDs

Questo esempio associa un documento di configurazione a un'istanza, utilizzando `instanceIDs`.

```

aws ssm create-association \
  --instance-id "i-0cb2b964d3e14fd9f" \
  --name "AWS-UpdateSSMAgent"

```

Output:

```

{
  "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"
    }
  ]
}

```

Per ulteriori informazioni, vedere [CreateAssociation AWS Systems Manager API Reference](#).

Esempio 2: associare un documento utilizzando obiettivi

Questo esempio associa un documento di configurazione a un'istanza, utilizzando obiettivi.

```

aws ssm create-association \
  --name "AWS-UpdateSSMAgent" \
  --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f"

```

Output:

```

{
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, vedere [CreateAssociation AWS Systems Manager API Reference](#).

Esempio 3: creare un'associazione che venga eseguita una sola volta

Questo esempio crea una nuova associazione che viene eseguita solo una volta nella data e ora specificate. Le associazioni create con una data passata o presente (nel momento in cui viene elaborata, la data è passata) vengono eseguite immediatamente.

```

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

```

Output:

```

{
  "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"
      }
    ]
  }
}

```

Per ulteriori informazioni, vedere [CreateAssociation AWS Systems Manager API Reference](#) o [Reference: Cron and rate expressions for Systems Manager](#) nella *AWS Systems Manager User Guide*.

- Per API i dettagli, vedere [CreateAssociation](#) in *AWS CLI Command Reference*.

## create-document

Il seguente esempio di codice mostra come utilizzare `create-document`.

### AWS CLI

Per creare un documento

L'`create-document` esempio seguente crea un documento Systems Manager.

```

aws ssm create-document \
  --content file://exampleDocument.yml \
  --name "Example" \
  --document-type "Automation" \
  --document-format YAML

```

## Output:

```
{
  "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": []
  }
}
```

Per ulteriori informazioni, vedere [Creazione di documenti di Systems Manager](#) nella Guida per l'utente di AWS Systems Manager.



- Per API i dettagli, vedere [CreateDocument](#) in AWS CLI Command Reference.

## create-maintenance-window

Il seguente esempio di codice mostra come utilizzare `create-maintenance-window`.

### AWS CLI

#### Esempio 1: creare una finestra di manutenzione

L'esempio seguente crea una nuova finestra di manutenzione che ogni cinque minuti per un massimo di due ore (se necessario), impedisce l'avvio di nuove attività entro un'ora dalla fine dell'esecuzione della finestra di manutenzione, consente destinazioni non associate (istanze che non sono state registrate nella finestra di manutenzione) e indica, tramite l'uso di tag personalizzati, che il suo creatore intende utilizzarla in un tutorial.

```
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"
```

#### Output:

```
{  
  "WindowId": "mw-0c50858d01EXAMPLE"  
}
```

#### Esempio 2: creare una finestra di manutenzione che venga eseguita una sola volta

L'esempio seguente crea una nuova finestra di manutenzione che viene eseguita solo una volta nella data e all'ora specificate.

```
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"
```

Output:

```
{
  "WindowId": "mw-01234567890abcdef"
}
```

Per ulteriori informazioni, vedere [Maintenance Windows](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [CreateMaintenanceWindow](#) in AWS CLI Command Reference.

## create-ops-item

Il seguente esempio di codice mostra come utilizzare `create-ops-item`.

### AWS CLI

Per creare un OpsItems

L'esempio seguente utilizza la `/aws/resources` key in `OperationalData` per creare una risorsa OpsItem correlata ad Amazon DynamoDB.

```
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"}], "Type": "SearchableString"}}' \
  --notifications Arn="arn:aws:sns:us-west-2:12345678:TestUser"
```

Output:

```
{
  "OpsItemId": "oi-1a2b3c4d5e6f"
}
```

Per ulteriori informazioni, vedere [Creating OpsItems](#) in the AWS Systems Manager User Guide.

- Per API i dettagli, vedere [CreateOpsItem](#) in AWS CLI Command Reference.

## create-patch-baseline

Il seguente esempio di codice mostra come utilizzare `create-patch-baseline`.

### AWS CLI

Esempio 1: creare una baseline di patch con approvazione automatica

L'`create-patch-baseline` seguente crea una linea di base delle patch per Windows Server che approva le patch per un ambiente di produzione sette giorni dopo il loro rilascio da parte di Microsoft.

```
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"
```

Output:

```
{
  "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

Esempio 2: creare una baseline di patch con una data limite di approvazione

L'`create-patch-baseline` seguente crea una patch di base per Windows Server che approva tutte le patch per un ambiente di produzione rilasciate entro il 7 luglio 2020.

```
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"
```

**Output:**

```
{
  "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

Esempio 3: creare una baseline di patch con regole di approvazione archiviate in un file JSON

L'create-patch-baselinesesempio seguente crea una patch baseline per Amazon Linux 2017.09 che approva le patch per un ambiente di produzione sette giorni dopo il loro rilascio, specifica le regole di approvazione per la patch baseline e specifica un repository personalizzato per le patch.

```
aws ssm create-patch-baseline \  
  --cli-input-json file://my-amazon-linux-approval-rules-and-repo.json
```

Contenuto di 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"
  }
]
}

```

Esempio 4: creare una linea di base delle patch che specifichi le patch approvate e rifiutate

L'`create-patch-baseline` seguente specifica esplicitamente le patch da approvare e rifiutare come eccezione alle regole di approvazione predefinite.

```

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"
```

Per ulteriori informazioni, vedere [Create a Custom Patch Baseline](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [CreatePatchBaseline](#) in AWS CLI Command Reference.

## create-resource-data-sync

Il seguente esempio di codice mostra come utilizzare `create-resource-data-sync`.

### AWS CLI

Per creare una sincronizzazione dei dati delle risorse

Questo esempio crea una sincronizzazione dei dati delle risorse. Se il comando va a buon fine, non viene generato output.

Comando:

```
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"
```

- Per API i dettagli, vedere [CreateResourceDataSync](#) in AWS CLI Command Reference.

## delete-activation

Il seguente esempio di codice mostra come utilizzare `delete-activation`.

### AWS CLI

Per eliminare l'attivazione di un'istanza gestita

L'esempio seguente elimina l'attivazione di un'istanza gestita.

```
aws ssm delete-activation \  
--activation-id "aa673477-d926-42c1-8757-1358cEXAMPLE"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Configurazione di AWS Systems Manager per ambienti ibridi](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DeleteActivation](#) in AWS CLI Command Reference.

## delete-association

Il seguente esempio di codice mostra come utilizzare `delete-association`.

### AWS CLI

Esempio 1: Per eliminare un'associazione utilizzando l'ID dell'associazione

L'`delete-association` esempio seguente elimina l'associazione per l'ID di associazione specificato. Se il comando va a buon fine, non viene generato output.

```
aws ssm delete-association \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Modifica e creazione di una nuova versione di un'associazione](#) nella Guida per l'utente di AWS Systems Manager.

Esempio 2: eliminare un'associazione

L'`delete-association` esempio seguente elimina l'associazione tra un'istanza e un documento. Se il comando va a buon fine, non viene generato output.

```
aws ssm delete-association \  
  --instance-id "i-1234567890abcdef0" \  
  --name "AWS-UpdateSSMAgent"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Utilizzo delle associazioni in Systems Manager](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DeleteAssociation](#) in AWS CLI Command Reference.

## delete-document

Il seguente esempio di codice mostra come utilizzare `delete-document`.

### AWS CLI

Per eliminare un documento

L'esempio seguente elimina un documento Systems Manager.

```
aws ssm delete-document \  
  --name "Example"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Creazione di documenti di Systems Manager](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DeleteDocument](#) in AWS CLI Command Reference.

## delete-inventory

Il seguente esempio di codice mostra come utilizzare `delete-inventory`.

### AWS CLI

Per eliminare un tipo di inventario personalizzato

Questo esempio elimina uno schema di inventario personalizzato.

Comando:

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-  
option "DeleteSchema"
```

Output:

```
{  
  "DeletionId": "d72ac9e8-1f60-4d40-b1c6-bf8c78c68c4d",  
  "TypeName": "Custom:RackInfo",  
  "DeletionSummary": {  
    "TotalCount": 1,  
  }  
}
```



```
    "RemainingCount": 1,
    "SummaryItems": [
      {
        "Version": "1.0",
        "Count": 1,
        "RemainingCount": 1
      }
    ]
  }
}
```

Per disabilitare un tipo di inventario personalizzato

Questo esempio disabilita uno schema di inventario personalizzato.

Comando:

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-  
option "DisableSchema"
```

Output:

```
{
  "DeletionId": "6961492a-8163-44ec-aa1e-923364dd0850",
  "TypeName": "Custom:RackInformation",
  "DeletionSummary": {
    "TotalCount": 0,
    "RemainingCount": 0,
    "SummaryItems": []
  }
}
```

- Per API i dettagli, vedere [DeleteInventory](#) in AWS CLI Command Reference.

## delete-maintenance-window

Il seguente esempio di codice mostra come utilizzare `delete-maintenance-window`.

AWS CLI

Per eliminare una finestra di manutenzione

Questo `delete-maintenance-window` esempio rimuove la finestra di manutenzione specificata.

```
aws ssm delete-maintenance-window \  
  --window-id "mw-1a2b3c4d5e6f7g8h9"
```

Output:

```
{  
  "WindowId": "mw-1a2b3c4d5e6f7g8h9"  
}
```

Per ulteriori informazioni, vedere [Eliminare una finestra di manutenzione \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DeleteMaintenanceWindow](#) in AWS CLI Command Reference.

## **delete-parameter**

Il seguente esempio di codice mostra come utilizzare `delete-parameter`.

AWS CLI

Per eliminare un parametro

L'`delete-parameter` esempio seguente elimina il singolo parametro specificato.

```
aws ssm delete-parameter \  
  --name "MyParameter"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Working with Parameter Store](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [DeleteParameter](#) in AWS CLI Command Reference.

## **delete-parameters**

Il seguente esempio di codice mostra come utilizzare `delete-parameters`.

## AWS CLI

Per eliminare un elenco di parametri

L'`delete-parameters` esempio seguente elimina i parametri specificati.

```
aws ssm delete-parameters \  
  --names "MyFirstParameter" "MySecondParameter" "MyInvalidParameterName"
```

Output:

```
{  
  "DeletedParameters": [  
    "MyFirstParameter",  
    "MySecondParameter"  
  ],  
  "InvalidParameters": [  
    "MyInvalidParameterName"  
  ]  
}
```

Per ulteriori informazioni, vedere [Working with Parameter Store](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [DeleteParameters](#) in AWS CLI Command Reference.

## `delete-patch-baseline`

Il seguente esempio di codice mostra come utilizzare `delete-patch-baseline`.

## AWS CLI

Per eliminare una patch baseline

L'`delete-patch-baseline` esempio seguente elimina la linea di base della patch specificata.

```
aws ssm delete-patch-baseline \  
  --baseline-id "pb-045f10b4f382baeda"
```

Output:

```
{
```

```
"BaselineId": "pb-045f10b4f382baeda"  
}
```

Per ulteriori informazioni, vedere [Update or Delete a Patch Baseline \(console\)](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [DeletePatchBaseline](#) in AWS CLI Command Reference.

## **delete-resource-data-sync**

Il seguente esempio di codice mostra come utilizzare `delete-resource-data-sync`.

### AWS CLI

Per eliminare una risorsa, la sincronizzazione dei dati

Questo esempio elimina una sincronizzazione dei dati delle risorse. Se il comando va a buon fine, non viene generato output.

Comando:

```
aws ssm delete-resource-data-sync --sync-name "ssm-resource-data-sync"
```

- Per API i dettagli, vedere [DeleteResourceDataSync](#) in AWS CLI Command Reference.

## **deregister-managed-instance**

Il seguente esempio di codice mostra come utilizzare `deregister-managed-instance`.

### AWS CLI

Per annullare la registrazione di un'istanza gestita

L'`deregister-managed-instance` seguente annulla la registrazione dell'istanza gestita specificata.

```
aws ssm deregister-managed-instance  
--instance-id "mi-08ab247cdfEXAMPLE"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Annullamento della registrazione delle istanze gestite in un ambiente ibrido](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DeregisterManagedInstance](#) in AWS CLI Command Reference.

## deregister-patch-baseline-for-patch-group

Il seguente esempio di codice mostra come utilizzare `deregister-patch-baseline-for-patch-group`.

### AWS CLI

Per annullare la registrazione di un gruppo di patch da una linea di base di patch

L'esempio seguente annulla la registrazione del gruppo di patch specificato dalla baseline di patch specificata.

```
aws ssm deregister-patch-baseline-for-patch-group \  
  --patch-group "Production" \  
  --baseline-id "pb-0ca44a362fEXAMPLE"
```

Output:

```
{  
  "PatchGroup": "Production",  
  "BaselineId": "pb-0ca44a362fEXAMPLE"  
}
```

Per ulteriori informazioni, vedere [Aggiungere un gruppo di patch a una patch baseline](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DeregisterPatchBaselineForPatchGroup](#) in AWS CLI Command Reference.

## deregister-target-from-maintenance-window

Il seguente esempio di codice mostra come utilizzare `deregister-target-from-maintenance-window`.

### AWS CLI

Per rimuovere un obiettivo da una finestra di manutenzione

L'`deregister-target-from-maintenance-window` seguente rimuove la destinazione specificata dalla finestra di manutenzione specificata.

```
aws ssm deregister-target-from-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --window-target-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
```

Output:

```
{  
  "WindowId": "mw-ab12cd34ef56gh78",  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

Per ulteriori informazioni, vedere [Update a Maintenance Window \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DeregisterTargetFromMaintenanceWindow](#) in AWS CLI Command Reference.

## **deregister-task-from-maintenance-window**

Il seguente esempio di codice mostra come utilizzare `deregister-task-from-maintenance-window`.

AWS CLI

Per rimuovere un'attività da una finestra di manutenzione

L'`deregister-task-from-maintenance-window` seguente rimuove l'attività specificata dalla finestra di manutenzione specificata.

```
aws ssm deregister-task-from-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --window-task-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c"
```

Output:

```
{  
  "WindowTaskId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c",  
  "WindowId": "mw-ab12cd34ef56gh78"  
}
```

```
}
```

Per ulteriori informazioni, vedere [Systems Manager Maintenance Windows Tutorials \(AWS CLI\)](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [DeregisterTaskFromMaintenanceWindow](#) in AWS CLI Command Reference.

## describe-activations

Il seguente esempio di codice mostra come utilizzare `describe-activations`.

### AWS CLI

Per descrivere le attivazioni

L'`describe-activations` esempio seguente elenca i dettagli sulle attivazioni nel tuo AWS account.

```
aws ssm describe-activations
```

Output:

```
{
  "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
  },
]
}

```

Per ulteriori informazioni, vedere [Passaggio 4: Creare un'attivazione di istanze gestite per un ambiente ibrido](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeActivations](#) in AWS CLI Command Reference.

## describe-association-execution-targets

Il seguente esempio di codice mostra come utilizzare `describe-association-execution-targets`.

### AWS CLI

Per ottenere dettagli sull'esecuzione di un'associazione

L'esempio seguente descrive l'esecuzione dell'associazione specificata.

```

aws ssm describe-association-execution-targets \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
  --execution-id "7abb6378-a4a5-4f10-8312-0123456789ab"

```

Output:

```

{
  "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"
      }
    }
  ]
}

```



```

    }
  ]
}

```

Per ulteriori informazioni, vedere [Visualizzazione della cronologia delle associazioni](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeAssociationExecutionTargets](#) in AWS CLI Command Reference.

## describe-association-executions

Il seguente esempio di codice mostra come utilizzare `describe-association-executions`.

### AWS CLI

Esempio 1: Per ottenere i dettagli di tutte le esecuzioni di un'associazione

L'`describe-association-executions` seguente descrive tutte le esecuzioni dell'associazione specificata.

```

aws ssm describe-association-executions \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"

```

Output:

```

{
  "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}"
    },
    ...
]
}

```

Per ulteriori informazioni, vedere [Visualizzazione della cronologia delle associazioni](#) nella Guida per l'utente di AWS Systems Manager.

Esempio 2: per ottenere i dettagli di tutte le esecuzioni di un'associazione dopo una data e un'ora specifiche

L'`describe-association-executions` esempio seguente descrive tutte le esecuzioni di un'associazione dopo la data e l'ora specificate.

```

aws ssm describe-association-executions \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
  --filters "Key=CreatedTime,Value=2019-02-18T16:00:00Z,Type=GREATER_THAN"

```

Output:

```

{
  "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}"
    },
    ...
  ]
}

```

```
]
}
```

Per ulteriori informazioni, vedere [Visualizzazione della cronologia delle associazioni](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeAssociationExecutions](#) in AWS CLI Command Reference.

## describe-association

Il seguente esempio di codice mostra come utilizzare `describe-association`.

### AWS CLI

Esempio 1: Per ottenere i dettagli di un'associazione

L'`describe-association`esempio seguente descrive l'associazione per l'ID di associazione specificato.

```
aws ssm describe-association \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Output:

```
{
  "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"
}
}
```

Per ulteriori informazioni, vedere [Modifica e creazione di una nuova versione di un'associazione](#) nella Guida per l'utente di AWS Systems Manager.

Esempio 2: per ottenere i dettagli di un'associazione per un'istanza e un documento specifici

L'`describe-association`esempio seguente descrive l'associazione tra un'istanza e un documento.

```
aws ssm describe-association \  
  --instance-id "i-1234567890abcdef0" \  
  --name "AWS-UpdateSSMAgent"
```

Output:

```
{  
  "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"  
      }  
    ]  
  }  
}
```

```
}
```

Per ulteriori informazioni, vedere [Modifica e creazione di una nuova versione di un'associazione](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeAssociation](#) in AWS CLI Command Reference.

## describe-automation-executions

Il seguente esempio di codice mostra come utilizzare `describe-automation-executions`.

### AWS CLI

Per descrivere un'esecuzione di automazione

L'`describe-automation-executions` seguente mostra i dettagli sull'esecuzione di un'automazione.

```
aws ssm describe-automation-executions \  
  --filters Key=ExecutionId,Values=73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

Output:

```
{  
  "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"
      }
    ]
  }

```

Per ulteriori informazioni, vedere [Running a Simple Automation Workflow](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [DescribeAutomationExecutions](#) in AWS CLI Command Reference.

## describe-automation-step-executions

Il seguente esempio di codice mostra come utilizzare `describe-automation-step-executions`.

### AWS CLI

Esempio 1: Per descrivere tutti i passaggi per l'esecuzione di un'automazione

L'`describe-automation-step-executions` esempio seguente mostra i dettagli sui passaggi di un'esecuzione di automazione.

```

aws ssm describe-automation-step-executions \
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE

```

Output:

```

{
  "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": {}
  }
]
}

```

Esempio 2: Per descrivere un passaggio specifico per l'esecuzione di un'automazione

L'`describe-automation-step-executions` esempio seguente mostra i dettagli su un passaggio specifico di un'esecuzione di automazione.

```

aws ssm describe-automation-step-executions \
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \
  --filters Key=StepExecutionId,Values=95e70479-cf20-4d80-8018-7e4e2EXAMPLE

```

Per ulteriori informazioni, vedere [Esecuzione di un flusso di lavoro di automazione passo dopo passo \(riga di comando\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeAutomationStepExecutions](#) in AWS CLI Command Reference.

## describe-available-patches

Il seguente esempio di codice mostra come utilizzare `describe-available-patches`.

### AWS CLI

Per ottenere le patch disponibili

L'`describe-available-patches` esempio seguente recupera i dettagli su tutte le patch disponibili per Windows Server 2019 che hanno una MSRC gravità critica.

```

aws ssm describe-available-patches \
  --
  filters "Key=PRODUCT,Values=WindowsServer2019" "Key=MSRC_SEVERITY,Values=Critical"

```

Output:

```

{
  "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"
    },
    ...
]
}

```

Per ottenere i dettagli di una patch specifica

L'`describe-available-patches`esempio seguente recupera i dettagli sulla patch specificata.

```
aws ssm describe-available-patches \
  --filters "Key=PATCH_ID,Values=KB4480979"
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Come funzionano le operazioni di Patch Manager](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeAvailablePatches](#) in AWS CLI Command Reference.

## describe-document-permission

Il seguente esempio di codice mostra come utilizzare `describe-document-permission`.

AWS CLI

Per descrivere le autorizzazioni dei documenti

L' `describe-document-permission` esempio seguente visualizza i dettagli delle autorizzazioni relative a un documento Systems Manager condiviso pubblicamente.

```
aws ssm describe-document-permission \  
  --name "Example" \  
  --permission-type "Share"
```

Output:

```
{  
  "AccountIds": [  
    "all"  
  ],  
  "AccountSharingInfoList": [  
    {  
      "AccountId": "all",  
      "SharedDocumentVersion": "$DEFAULT"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Share a Systems Manager Document](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [DescribeDocumentPermission](#) in AWS CLI Command Reference.

## `describe-document`

Il seguente esempio di codice mostra come utilizzare `describe-document`.

### AWS CLI

Per visualizzare i dettagli di un documento

L' `describe-document` esempio seguente mostra i dettagli su un documento Systems Manager nel tuo AWS account.

```
aws ssm describe-document \  
  --name "Example"
```

Output:

```
{
  "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": []
  }
}
```

Per ulteriori informazioni, vedere [Creazione di documenti di Systems Manager](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeDocument](#) in AWS CLI Command Reference.

## describe-effective-instance-associations

Il seguente esempio di codice mostra come utilizzare `describe-effective-instance-associations`.

### AWS CLI

Per ottenere dettagli sulle associazioni efficaci per un'istanza

L'`describe-effective-instance-associations`esempio seguente recupera i dettagli sulle associazioni efficaci per un'istanza.

Comando:

```
aws ssm describe-effective-instance-associations --instance-id "i-1234567890abcdef0"
```

Output:

```
{
  "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    },\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  \"AssociationVersion\": \"1\"
    }
  ]
}
```

```
]
}
```

- Per API i dettagli, vedete [DescribeEffectiveInstanceAssociations](#) in AWS CLI Command Reference.

## describe-effective-patches-for-patch-baseline

Il seguente esempio di codice mostra come utilizzare `describe-effective-patches-for-patch-baseline`.

### AWS CLI

Esempio 1: per ottenere tutte le patch definite da una linea di base di patch personalizzata

L'`describe-effective-patches-for-patch-baseline` esempio seguente restituisce le patch definite da una linea di base di patch personalizzata nell'account corrente. AWS Si noti che per una baseline personalizzata, è richiesto solo l'ID. `--baseline-id`

```
aws ssm describe-effective-patches-for-patch-baseline \
  --baseline-id "pb-08b654cf9b9681f04"
```

Output:

```
{
  "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--"
}

```

Esempio 2: Per ottenere tutte le patch definite da una baseline di patch AWS gestite

L' `aws ssm describe-effective-patches-for-patch-baseline` seguente restituisce le patch definite da una baseline di patch AWS gestite. Si noti che per una baseline AWS gestita, è richiesta la ARN baseline completa per `--baseline-id`

```
aws ssm describe-effective-patches-for-patch-baseline \  
  --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/  
pb-020d361a05defe4ed"
```

Vedi l'esempio 1 per un esempio di output.

Per ulteriori informazioni, vedere [Come vengono selezionate le patch di sicurezza](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeEffectivePatchesForPatchBaseline](#) in AWS CLI Command Reference.

## `describe-instance-associations-status`

Il seguente esempio di codice mostra come utilizzare `describe-instance-associations-status`.

### AWS CLI

Per descrivere lo stato delle associazioni di un'istanza

Questo esempio mostra i dettagli delle associazioni per un'istanza.

Comando:

```
aws ssm describe-instance-associations-status --instance-id "i-1234567890abcdef0"
```

Output:

```
{  
  "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"
  }
]
}

```

- Per API i dettagli, vedere [DescribeInstanceAssociationsStatus](#) in AWS CLI Command Reference.

## describe-instance-information

Il seguente esempio di codice mostra come utilizzare `describe-instance-information`.

### AWS CLI

Esempio 1: Per descrivere le informazioni sulle istanze gestite

L'`describe-instance-information`esempio seguente recupera i dettagli di ciascuna delle istanze gestite.

```
aws ssm describe-instance-information
```

Esempio 2: Per descrivere le informazioni su una specifica istanza gestita

L'`describe-instance-information`esempio seguente mostra i dettagli dell'istanza gestita `i-028ea792daEXAMPLE`.

```
aws ssm describe-instance-information \
```

```
--filters "Key=InstanceIds,Values=i-028ea792daEXAMPLE"
```

Esempio 3: descrivere le informazioni sulle istanze gestite con una chiave di tag specifica

L'`describe-instance-information`esempio seguente mostra i dettagli delle istanze gestite con la chiave tag. DEV

```
aws ssm describe-instance-information \  
--filters "Key=tag-key,Values=DEV"
```

Output:

```
{  
  "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  
        }  
      }  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Managed Instances](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [DescribeInstanceInformation](#) in AWS CLI Command Reference.

## describe-instance-patch-states-for-patch-group

Il seguente esempio di codice mostra come utilizzare `describe-instance-patch-states-for-patch-group`.

### AWS CLI

Esempio 1: per ottenere gli stati delle istanze per un gruppo di patch

L'`describe-instance-patch-states-for-patch-group` seguente recupera i dettagli sugli stati di riepilogo delle patch per istanza per il gruppo di patch specificato.

```
aws ssm describe-instance-patch-states-for-patch-group \
  --patch-group "Production"
```

Output:

```
{
  "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
  }
]
}

```

Esempio 2: Per ottenere gli stati delle istanze per un gruppo di patch con più di cinque patch mancanti

L'operazione `describe-instance-patch-states-for-patch-group` seguente recupera i dettagli sugli stati di riepilogo delle patch per il gruppo di patch specificato per le istanze con più di cinque patch mancanti.

```

aws ssm describe-instance-patch-states-for-patch-group \
  --filters Key=MissingCount,Type=GreaterThan,Values=5 \
  --patch-group "Production"

```

Output:

```

{
  "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
  }
]
}

```

Esempio 3: Per ottenere gli stati delle istanze per un gruppo di patch con meno di dieci istanze che richiedono un riavvio

L'operazione `describe-instance-patch-states-for-patch-group` seguente recupera i dettagli sugli stati di riepilogo delle patch per il gruppo di patch specificato per le istanze con meno di dieci istanze che richiedono un riavvio.

```

aws ssm describe-instance-patch-states-for-patch-group \
  --filters Key=InstalledPendingRebootCount,Type=LessThan,Values=10 \
  --patch-group "Production"

```

Output:

```

{
  "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
  }
]
}

```

Per ulteriori informazioni, vedere [Comprensione dei valori dello stato di conformità delle patch](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [DescribeInstancePatchStatesForPatchGroup](#) in AWS CLI Command Reference.

## describe-instance-patch-states

Il seguente esempio di codice mostra come utilizzare `describe-instance-patch-states`.

### AWS CLI

Per ottenere gli stati di riepilogo delle patch per le istanze

Questo `describe-instance-patch-states` esempio ottiene gli stati di riepilogo delle patch per un'istanza.

```
aws ssm describe-instance-patch-states \
  --instance-ids "i-1234567890abcdef0"
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, vedere [Informazioni sulla conformità delle patch](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeInstancePatchStates](#) in AWS CLI Command Reference.

## describe-instance-patches

Il seguente esempio di codice mostra come utilizzare `describe-instance-patches`.

### AWS CLI

Esempio 1: per ottenere i dettagli sullo stato della patch per un'istanza

L'`describe-instance-patches` esempio seguente recupera i dettagli sulle patch per l'istanza specificata.

```
aws ssm describe-instance-patches \
  --instance-id "i-1234567890abcdef0"
```

## Output:

```
{
  "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--"
}
```

Esempio 2: per ottenere un elenco di patch nello stato Missing per un'istanza

L'describe-instance-patchesesempio seguente recupera informazioni sulle patch che si trovano nello stato Missing per l'istanza specificata.

```
aws ssm describe-instance-patches \
  --instance-id "i-1234567890abcdef0" \
  --filters Key=State,Values=Missing
```

## Output:

```
{
  "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--"
}

```

Per ulteriori informazioni, vedere Informazioni [sugli stati di conformità delle patch](#) nella Guida per l'utente di AWS Systems Manager.

Esempio 3: Per ottenere un elenco delle patch installate dopo quella specificata `InstalledTime` per un'istanza

L'`describe-instance-patches` seguente recupera le informazioni sulle patch installate a partire da un determinato periodo di tempo per l'istanza specificata combinando l'uso di `and`. `--filters` `--query`

```

aws ssm describe-instance-patches \
  --instance-id "i-1234567890abcdef0" \
  --filters Key=State,Values=Installed \
  --query "Patches[?InstalledTime >= `2023-01-01T16:00:00`]"

```

Output:

```

{
  "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--"
}

```

- Per API i dettagli, vedere [DescribeInstancePatches](#) in AWS CLI Command Reference.

## describe-inventory-deletions

Il seguente esempio di codice mostra come utilizzare `describe-inventory-deletions`.

### AWS CLI

Per ottenere le eliminazioni dell'inventario

Questo esempio recupera i dettagli per le operazioni di eliminazione dell'inventario.

Comando:

```
aws ssm describe-inventory-deletions
```

Output:

```
{
  "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
}
]
}

```

Per ottenere i dettagli di una specifica eliminazione dell'inventario

Questo esempio recupera i dettagli per una specifica operazione di eliminazione dell'inventario.

Comando:

```
aws ssm describe-inventory-deletions --deletion-id "d72ac9e8-1f60-4d40-b1c6-987654321c4d"
```

Output:

```

{
  "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
}

```

```

    }
  ]
}

```

- Per API i dettagli, vedere [DescribeInventoryDeletions](#) in AWS CLI Command Reference.

## describe-maintenance-window-execution-task-invocations

Il seguente esempio di codice mostra come utilizzare `describe-maintenance-window-execution-task-invocations`.

### AWS CLI

Per eseguire le chiamate di attività specifiche per una finestra di manutenzione, l'esecuzione dell'operazione.

L'`describe-maintenance-window-execution-task-invocations` esempio seguente elenca le chiamate per l'attività specificata eseguite come parte dell'esecuzione della finestra di manutenzione specificata.

```

aws ssm describe-maintenance-window-execution-task-invocations \
  --window-execution-id "518d5565-5969-4cca-8f0e-da3b2a638355" \
  --task-id "ac0c6ae1-daa3-4a89-832e-d384503b6586"

```

### Output:

```

{
  "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"
    }
  ]
}

```

Per ulteriori informazioni, vedere [View Information About Tasks and Task Executions \(AWS CLI\)](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [DescribeMaintenanceWindowExecutionTaskInvocations](#) in AWS CLI Command Reference.

## **describe-maintenance-window-execution-tasks**

Il seguente esempio di codice mostra come utilizzare `describe-maintenance-window-execution-tasks`.

### AWS CLI

Per elencare tutte le attività associate all'esecuzione di una finestra di manutenzione

L'ssm `describe-maintenance-window-execution-tasks` esempio seguente elenca le attività associate all'esecuzione della finestra di manutenzione specificata.

```
aws ssm describe-maintenance-window-execution-tasks \
  --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [View Information About Tasks and Task Executions \(AWS CLI\)](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [DescribeMaintenanceWindowExecutionTasks](#) in AWS CLI Command Reference.

## describe-maintenance-window-executions

Il seguente esempio di codice mostra come utilizzare `describe-maintenance-window-executions`.

### AWS CLI

Esempio 1: Per elencare tutte le esecuzioni per una finestra di manutenzione

L'esempio seguente elenca tutte le esecuzioni per la finestra di manutenzione specificata.

```
aws ssm describe-maintenance-window-executions \
  --window-id "mw-ab12cd34eEXAMPLE"
```

Output:

```
{
  "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"
    }
  ]
}
```

Esempio 2: Per elencare tutte le esecuzioni relative a una finestra di manutenzione prima di una data specificata

L'`describe-maintenance-window-executions` seguente elenca tutte le esecuzioni per la finestra di manutenzione specificata prima della data specificata.

```
aws ssm describe-maintenance-window-executions \  
  --window-id "mw-ab12cd34eEXAMPLE" \  
  --filters "Key=ExecutedBefore,Values=2021-08-03T00:00:00Z"
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Esempio 3: Per elencare tutte le esecuzioni per una finestra di manutenzione dopo una data specificata

L'`describe-maintenance-window-executions` seguente elenca tutte le esecuzioni per la finestra di manutenzione specificata dopo la data specificata.

```
aws ssm describe-maintenance-window-executions \  
  --window-id "mw-ab12cd34eEXAMPLE" \  
  --filters "Key=ExecutedAfter,Values=2021-08-04T00:00:00Z"
```

Output:

```
{  
  "WindowExecutions": [  
    {  
      "WindowId": "mw-ab12cd34eEXAMPLE",  
      "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",  

```

```
    "Status": "IN_PROGRESS",
    "StartTime": "2021-08-04T11:00:00.000000-07:00"
  }
]
}
```

Per ulteriori informazioni, vedere [Visualizza informazioni sulle attività e sulle esecuzioni delle attività \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeMaintenanceWindowExecutions](#) in AWS CLI Command Reference.

## describe-maintenance-window-schedule

Il seguente esempio di codice mostra come utilizzare `describe-maintenance-window-schedule`.

### AWS CLI

Esempio 1: Per elencare le esecuzioni imminenti per una finestra di manutenzione

L'esempio seguente elenca tutte le esecuzioni imminenti per la finestra di manutenzione specificata.

```
aws ssm describe-maintenance-window-schedule \
  --window-id mw-ab12cd34eEXAMPLE
```

Output:

```
{
  "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"
    },
    ...
  ]
}
```



```
]
}
```

Esempio 2: Per elencare tutte le esecuzioni imminenti per una finestra di manutenzione prima di una data specificata

L'`describe-maintenance-window-schedule` seguente elenca tutte le esecuzioni imminenti per la finestra di manutenzione specificata che si verificano prima della data specificata.

```
aws ssm describe-maintenance-window-schedule \
  --window-id mw-0ecb1226dd7b2e9a6 \
  --filters "Key=ScheduledBefore,Values=2020-02-15T06:00:00Z"
```

Esempio 3: Per elencare tutte le esecuzioni imminenti per una finestra di manutenzione dopo una data specificata

L'`describe-maintenance-window-schedule` seguente elenca tutte le esecuzioni imminenti per la finestra di manutenzione specificata che si verificano dopo la data specificata.

```
aws ssm describe-maintenance-window-schedule \
  --window-id mw-0ecb1226dd7b2e9a6 \
  --filters "Key=ScheduledAfter,Values=2020-02-15T06:00:00Z"
```

Per ulteriori informazioni, vedere [Visualizza informazioni sulla manutenzione di Windows \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeMaintenanceWindowSchedule](#) in AWS CLI Command Reference.

## **describe-maintenance-window-targets**

Il seguente esempio di codice mostra come utilizzare `describe-maintenance-window-targets`.

### AWS CLI

Esempio 1: Per elencare tutti gli obiettivi per una finestra di manutenzione

L'`describe-maintenance-window-targets` seguente elenca tutti gli obiettivi per una finestra di manutenzione.

```
aws ssm describe-maintenance-window-targets \
```

```
--window-id "mw-06cf17cbefEXAMPLE"
```

Output:

```
{
  "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"
    }
  ]
}
```

Esempio 2: elencare tutti gli obiettivi per una finestra di manutenzione che corrispondono a un valore informativo specifico sul proprietario

Questo `describe-maintenance-window-targets` esempio elenca tutte le destinazioni per una finestra di manutenzione con un valore specifico.

```
aws ssm describe-maintenance-window-targets \
  --window-id "mw-0ecb1226ddEXAMPLE" \
  --filters "Key=OwnerInformation,Values=CostCenter1"
```

Output:

```
{
  "Targets": [
    {
      "WindowId": "mw-0ecb1226ddEXAMPLE",
      "WindowTargetId": "da89dcc3-7f9c-481d-ba2b-edcb7d0057f9",
      "ResourceType": "INSTANCE",
      "Targets": [
        {
          "Key": "tag:Environment",
          "Values": [
            "Prod"
          ]
        }
      ],
      "OwnerInformation": "CostCenter1",
      "Name": "ProdTarget1"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Visualizza informazioni sulla manutenzione di Windows \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeMaintenanceWindowTargets](#) in AWS CLI Command Reference.

## describe-maintenance-window-tasks

Il seguente esempio di codice mostra come utilizzare `describe-maintenance-window-tasks`.

AWS CLI

Esempio 1: Per elencare tutte le attività per una finestra di manutenzione

L'`describe-maintenance-window-tasks`esempio seguente elenca tutte le attività per la finestra di manutenzione specificata.

```
aws ssm describe-maintenance-window-tasks \
  --window-id "mw-06cf17cbefEXAMPLE"
```

Output:

```
{
  "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"
        ]
      }
    ]
  }
]
}

```

Esempio 2: elencare tutte le attività per una finestra di manutenzione che richiama il documento di comando AWS- RunPowerShellScript

L'output seguente elenca tutte le attività per la finestra di manutenzione specificata che richiama il documento di comando. AWS- RunPowerShellScript

```

aws ssm describe-maintenance-window-tasks \
  --window-id "mw-ab12cd34eEXAMPLE" \
  --filters "Key=TaskArn,Values=AWS-RunPowerShellScript"

```

Output:

```

{
  "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"
    }
]
}

```

Esempio 3: per elencare tutte le attività per una finestra di manutenzione con una priorità di 3

L'output seguente elenca tutte le attività per la finestra di manutenzione specificata che hanno un valore Priority di 3.

```

aws ssm describe-maintenance-window-tasks \
  --window-id "mw-ab12cd34eEXAMPLE" \
  --filters "Key=Priority,Values=3"

```

Output:

```

{
  "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"
    }
  ]
}

```

Esempio 4: Per elencare tutte le attività per una finestra di manutenzione con priorità pari a 1 e utilizzare Esegui comando

Questo `describe-maintenance-window-tasks` esempio elenca tutte le attività per la finestra di manutenzione specificata che hanno un valore `Priority` di 1 e un utilizzo `Run Command`.

```

aws ssm describe-maintenance-window-tasks \
  --window-id "mw-ab12cd34eEXAMPLE" \
  --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"

```

Output:

```

{
  "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"
  }
]
}

```

Per ulteriori informazioni, vedere [Visualizza informazioni sulle finestre di manutenzione \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeMaintenanceWindowTasks](#) in AWS CLI Command Reference.

## describe-maintenance-windows-for-target

Il seguente esempio di codice mostra come utilizzare `describe-maintenance-windows-for-target`.

### AWS CLI

Per elencare tutte le finestre di manutenzione associate a un'istanza specifica

L'`describe-maintenance-windows-for-target` esempio seguente elenca le finestre di manutenzione con obiettivi o attività associati all'istanza specificata.

```

aws ssm describe-maintenance-windows-for-target \
  --targets Key=InstanceIds,Values=i-1234567890EXAMPLE \
  --resource-type INSTANCE

```

Output:



```
{
  "WindowIdentities": [
    {
      "WindowId": "mw-0c5ed765acEXAMPLE",
      "Name": "My-First-Maintenance-Window"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Visualizza informazioni sulla manutenzione di Windows \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeMaintenanceWindowsForTarget](#) in AWS CLI Command Reference.

## describe-maintenance-windows

Il seguente esempio di codice mostra come utilizzare `describe-maintenance-windows`.

### AWS CLI

Esempio 1: per elencare tutte le finestre di manutenzione

L'`describe-maintenance-windows` seguente elenca tutte le finestre di manutenzione del tuo AWS account nella regione corrente.

```
aws ssm describe-maintenance-windows
```

Output:

```
{
  "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"
    },
]
}

```

Esempio 2: per elencare tutte le finestre di manutenzione abilitate

L'`describe-maintenance-windows` esempio seguente elenca tutte le finestre di manutenzione abilitate.

```

aws ssm describe-maintenance-windows \
  --filters "Key=Enabled,Values=true"

```

Esempio 3: Per elencare le finestre di manutenzione che corrispondono a un nome specifico

Questo `describe-maintenance-windows` esempio elenca tutte le finestre di manutenzione con il nome specificato.

```

aws ssm describe-maintenance-windows \
  --filters "Key=Name,Values=MyMaintenanceWindow"

```

Per ulteriori informazioni, vedere [Visualizza informazioni sulla manutenzione di Windows \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeMaintenanceWindows](#) in AWS CLI Command Reference.

## **describe-ops-items**

Il seguente esempio di codice mostra come utilizzare `describe-ops-items`.

### AWS CLI

Per elencare un set di OpsItems

L'`describe-ops-items` esempio seguente mostra un elenco di tutti i dati aperti OpsItems nel tuo AWS account.

```
aws ssm describe-ops-items \
  --ops-item-filters "Key=Status,Values=Open,Operator=Equal"
```

Output:

```
{
  "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::1112223233444: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"
    }
]
}

```

Per ulteriori informazioni, vedere [Working with OpsItems](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [DescribeOpsItems](#) in AWS CLI Command Reference.

## describe-parameters

Il seguente esempio di codice mostra come utilizzare `describe-parameters`.

### AWS CLI

Esempio 1: Per elencare tutti i parametri

L'`describe-parameters` seguente elenca tutti i parametri dell' AWS account corrente e della regione.

```
aws ssm describe-parameters
```

Output:

```

{
  "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": []
    }
]
}

```

Esempio 2: per elencare tutti i parametri che corrispondono a metadati specifici

Questo `describe-parameters` esempio elenca tutti i parametri che corrispondono a un filtro.

```
aws ssm describe-parameters --filters «Key=Type, Values=» StringList
```

Output:

```

{
  "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": []
    }
  ]
}

```

Per ulteriori informazioni, vedere [Searching for Systems Manager Parameters](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [DescribeParameters](#) in AWS CLI Command Reference.

## describe-patch-baselines

Il seguente esempio di codice mostra come utilizzare `describe-patch-baselines`.

### AWS CLI

Esempio 1: per elencare tutte le linee di base delle patch

L'`describe-patch-baselines` seguente recupera i dettagli di tutte le patch di base del tuo account nella regione corrente.

```
aws ssm describe-patch-baselines
```

Output:

```
{
  "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"
    }
]
}

```

Esempio 2: Per elencare tutte le linee di base delle patch fornite da AWS

L'output di `aws ssm describe-patch-baselines` seguente elenca tutte le linee di base delle patch fornite da AWS.

```

aws ssm describe-patch-baselines \
  --filters "Key=OWNER,Values=[AWS]"

```

Esempio 3: Per elencare tutte le linee di base delle patch di cui sei proprietario

L'output di `aws ssm describe-patch-baselines` seguente elenca tutte le linee base di patch personalizzate create nel tuo account nella regione corrente.

```

aws ssm describe-patch-baselines \
  --filters "Key=OWNER,Values=[Self]"

```

Per ulteriori informazioni, vedere [Informazioni sulle patch di base predefinite e personalizzate nella Guida per l'utente](#) di AWS Systems Manager.

- Per API i dettagli, vedere [DescribePatchBaselines](#) in AWS CLI Command Reference.

## describe-patch-group-state

Il seguente esempio di codice mostra come utilizzare `describe-patch-group-state`.

AWS CLI

Per ottenere lo stato di un gruppo di patch

L'output di `aws ssm describe-patch-group-state` seguente recupera il riepilogo di alto livello sulla conformità delle patch per un gruppo di patch.

```

aws ssm describe-patch-group-state \
  --patch-group "Production"

```



**Output:**

```
{
  "Instances": 21,
  "InstancesWithCriticalNonCompliantPatches": 1,
  "InstancesWithFailedPatches": 2,
  "InstancesWithInstalledOtherPatches": 3,
  "InstancesWithInstalledPatches": 21,
  "InstancesWithInstalledPendingRebootPatches": 2,
  "InstancesWithInstalledRejectedPatches": 1,
  "InstancesWithMissingPatches": 3,
  "InstancesWithNotApplicablePatches": 4,
  "InstancesWithOtherNonCompliantPatches": 1,
  "InstancesWithSecurityNonCompliantPatches": 1,
  "InstancesWithUnreportedNotApplicablePatches": 2
}
```

Per ulteriori informazioni, vedere Informazioni sui gruppi di patch < <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-patchgroups.html> >\_\_ e Informazioni sui valori [dello stato di conformità delle](#) patch nella Systems Manager User Guide.AWS

- Per i API AWS CLI dettagli, [DescribePatchGroupState](#)vedere in Command Reference.

**describe-patch-groups**

Il seguente esempio di codice mostra come utilizzare `describe-patch-groups`.

**AWS CLI**

Per visualizzare le registrazioni dei gruppi di patch

L'`describe-patch-groups`esempio seguente elenca le registrazioni dei gruppi di patch.

```
aws ssm describe-patch-groups
```

**Output:**

```
{
  "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
    }
  },
  ...
]
}

```

Per ulteriori informazioni, vedere [Create a Patch Group < https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>](https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html) e [Aggiungere un gruppo di patch a una Patch Baseline nella Systems Manager AWS User Guide](#).

- Per i API dettagli, vedere in Command Reference. [DescribePatchGroups](#) AWS CLI

## describe-patch-properties

Il seguente esempio di codice mostra come utilizzare `describe-patch-properties`.

### AWS CLI

Per elencare la disponibilità delle patch di Amazon Linux

L'esempio seguente mostra un elenco dei prodotti Amazon Linux per i quali le patch sono disponibili nel tuo AWS account.

```

aws ssm describe-patch-properties \
  --operating-system AMAZON_LINUX \
  --property PRODUCT

```

## Output:

```
{
  "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"
    }
  ]
}
```

```
}
```

Per ulteriori informazioni, vedere [Informazioni sulle patch di base nella Guida per l'utente di AWS Systems Manager](#).

- Per API i dettagli, vedere [DescribePatchProperties](#) in AWS CLI Command Reference.

## describe-sessions

Il seguente esempio di codice mostra come utilizzare `describe-sessions`.

### AWS CLI

Esempio 1: per elencare tutte le sessioni attive di Session Manager

Questo `describe-sessions` esempio recupera un elenco delle sessioni attive create più di recente (sessioni connesse e disconnesse) negli ultimi 30 giorni che sono state avviate dall'utente specificato. Questo comando restituisce solo i risultati per le connessioni alle destinazioni avviate utilizzando Session Manager. Non elenca le connessioni effettuate con altri mezzi, come Remote Desktop Connections o SSH.

```
aws ssm describe-sessions \  
  --state "Active" \  
  --filters "key=Owner,value=arn:aws:sts::123456789012:assumed-role/Administrator/  
Shirley-Rodriguez"
```

Output:

```
{  
  "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": {}
    }
]
}

```

## Esempio 2: Per elencare tutte le sessioni di Session Manager terminate

Questo `describe-sessions` esempio recupera un elenco delle sessioni terminate più di recente negli ultimi 30 giorni per tutti gli utenti.

```

aws ssm describe-sessions \
  --state "History"

```

### Output:

```

{
  "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--"
}

```

Per ulteriori informazioni, vedere [Visualizza la cronologia delle sessioni](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [DescribeSessions](#) in AWS CLI Command Reference.

## **disassociate-ops-item-related-item**

Il seguente esempio di codice mostra come utilizzare `disassociate-ops-item-related-item`.

### AWS CLI

Per eliminare un'associazione di elementi correlati

L'`disassociate-ops-item-related-item` esempio seguente elimina l'associazione tra `OpsItem` e un elemento correlato.

```
aws ssm disassociate-ops-item-related-item \  
  --ops-item-id "oi-f99f2EXAMPLE" \  
  --association-id "e2036148-cccb-490e-ac2a-390e5EXAMPLE"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Utilizzo degli incidenti di Incident Manager OpsCenter nella Guida per l'utente di AWS Systems Manager](#).

- Per API i dettagli, vedere [DisassociateOpsItemRelatedItem](#) in AWS CLI Command Reference.

## **get-automation-execution**

Il seguente esempio di codice mostra come utilizzare `get-automation-execution`.

### AWS CLI

Per visualizzare i dettagli sull'esecuzione di un'automazione

L'`get-automation-execution` esempio seguente visualizza informazioni dettagliate sull'esecuzione di un'automazione.

```
aws ssm get-automation-execution \  
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

## Output:

```
{
  "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
    }
  }
}
```

Per ulteriori informazioni, vedere [Procedura dettagliata: Patch a Linux AMI \(AWS CLI\)](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [GetAutomationExecution](#) in AWS CLI Command Reference.

## get-calendar-state

Il seguente esempio di codice mostra come utilizzare `get-calendar-state`.

### AWS CLI

Esempio 1: Per ottenere lo stato corrente di un calendario delle modifiche

Questo `get-calendar-state` esempio restituisce lo stato di un calendario all'ora corrente. Poiché l'esempio non specifica un'ora, viene riportato lo stato corrente del calendario.

```
aws ssm get-calendar-state \
  --calendar-names "MyCalendar"
```

Output:

```
{
  "State": "OPEN",
  "AtTime": "2020-02-19T22:28:51Z",
  "NextTransitionTime": "2020-02-24T21:15:19Z"
}
```

Esempio 2: per ottenere lo stato di un calendario delle modifiche a un'ora specificata

Questo `get-calendar-state` esempio restituisce lo stato di un calendario all'ora specificata.

```
aws ssm get-calendar-state \
  --calendar-names "MyCalendar" \
```



```
--at-time "2020-07-19T21:15:19Z"
```

Output:

```
{
  "State": "CLOSED",
  "AtTime": "2020-07-19T21:15:19Z"
}
```

Per ulteriori informazioni, vedere [Get the State of the Change Calendar](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [GetCalendarState](#) in AWS CLI Command Reference.

## get-command-invocation

Il seguente esempio di codice mostra come utilizzare `get-command-invocation`.

AWS CLI

Per visualizzare i dettagli di una chiamata di comando

L'`get-command-invocation` esempio seguente elenca tutte le chiamate del comando specificato sull'istanza specificata.

```
aws ssm get-command-invocation \
  --command-id "ef7fd8-9b57-4151-a15c-db9a12345678" \
  --instance-id "i-1234567890abcdef0"
```

Output:

```
{
  "CommandId": "ef7fd8-9b57-4151-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
    }
}

```

Per ulteriori informazioni, vedere [Understanding Command Statuses](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [GetCommandInvocation](#) in AWS CLI Command Reference.

## get-connection-status

Il seguente esempio di codice mostra come utilizzare `get-connection-status`.

### AWS CLI

Per visualizzare lo stato della connessione di un'istanza gestita

Questo `get-connection-status` esempio restituisce lo stato della connessione dell'istanza gestita specificata.

```

aws ssm get-connection-status \
  --target i-1234567890abcdef0

```

Output:

```

{
  "Target": "i-1234567890abcdef0",
  "Status": "connected"
}

```

```
}
```

- Per API i dettagli, vedere [GetConnectionStatus](#) in AWS CLI Command Reference.

## get-default-patch-baseline

Il seguente esempio di codice mostra come utilizzare `get-default-patch-baseline`.

### AWS CLI

Esempio 1: Per visualizzare la baseline predefinita delle patch di Windows

L'`get-default-patch-baseline`esempio seguente recupera i dettagli per la patch di base predefinita per Windows Server.

```
aws ssm get-default-patch-baseline
```

Output:

```
{
  "BaselineId": "pb-0713accee01612345",
  "OperatingSystem": "WINDOWS"
}
```

Esempio 2: per visualizzare la linea di base delle patch predefinita per Amazon Linux

L'`get-default-patch-baseline`esempio seguente recupera i dettagli per la patch di base predefinita per Amazon Linux.

```
aws ssm get-default-patch-baseline \
  --operating-system AMAZON_LINUX
```

Output:

```
{
  "BaselineId": "pb-047c6eb9c8fc12345",
  "OperatingSystem": "AMAZON_LINUX"
}
```

Per ulteriori informazioni, vedere Informazioni sulle patch di base predefinite e personalizzate < <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-baselines.html>>\_\_ e Impostazione di una baseline di [patch esistente](#) come predefinita nella Systems Manager User Guide.AWS

- Per i API dettagli, [GetDefaultPatchBaseline](#)vedere AWS CLI in Command Reference.

## get-deployable-patch-snapshot-for-instance

Il seguente esempio di codice mostra come utilizzare `get-deployable-patch-snapshot-for-instance`.

### AWS CLI

Per recuperare l'istantanea corrente per la patch baseline, un'istanza utilizza

L'`get-deployable-patch-snapshot-for-instance`esempio seguente recupera i dettagli dello snapshot corrente per la baseline di patch specificata utilizzata da un'istanza. Questo comando deve essere eseguito dall'istanza utilizzando le credenziali dell'istanza. Per assicurarti che utilizzi le credenziali dell'istanza, esegui `aws configure` e specifica solo la regione dell'istanza. Lascia i `Secret Key` campi `Access Key` and vuoti.

Suggerimento: Utilizzare `uuidgen` per generare un `snapshot-id`.

```
aws ssm get-deployable-patch-snapshot-for-instance \
  --instance-id "i-1234567890abcdef0" \
  --snapshot-id "521c3536-930c-4aa9-950e-01234567abcd"
```

Output:

```
{
  "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=AKIAJ5C56P35AEBRX2Q0%2F20190215%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Signature=efaaaf6e3878e77f48a6697e015efdbda9c426b09c5822055075c062f6ad2149"
```

```
}

```

Per ulteriori informazioni, vedere [Parameter name: Snapshot ID](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [GetDeployablePatchSnapshotForInstance](#) in AWS CLI Command Reference.

## get-document

Il seguente esempio di codice mostra come utilizzare `get-document`.

### AWS CLI

Per ottenere il contenuto del documento

L'`get-document`esempio seguente visualizza il contenuto di un documento Systems Manager.

```
aws ssm get-document \
  --name "AWS-RunShellScript"
```

Output:

```
{
  "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\n  \"minItems\": 1, \n    \"displayType\": \"textarea\" \n  }, \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\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  }, \n  \"runtimeConfig\": {\n    \"aws:runShellScript\": {\n\n    \"properties\": [\n      {\n        \"id\":
```

```

\ "0.aws:runShellScript", \n
    \ "runCommand": \ "{ commands }", \n
    \ "workingDirectory": \ "{ workingDirectory }", \n
    \ "timeoutSeconds": \ "{ executionTimeout }", \n
  ] \n
  } \n
  } \n
  } \n
  "DocumentType": "Command",
  "DocumentFormat": "JSON"
}

```

Per ulteriori informazioni, vedere [AWS i documenti di Systems Manager](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [GetDocument](#) in AWS CLI Command Reference.

## get-inventory-schema

Il seguente esempio di codice mostra come utilizzare `get-inventory-schema`.

### AWS CLI

Per visualizzare lo schema dell'inventario

Questo esempio restituisce un elenco di nomi dei tipi di inventario per l'account.

Comando:

```
aws ssm get-inventory-schema
```

Output:

```

{
  "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--"
}

```

Per visualizzare lo schema di inventario per un tipo di inventario specifico

Questo esempio restituisce lo schema di inventario per un tipo di AWS inventario AWS dei componenti.

Comando:

```
aws ssm get-inventory-schema --type-name "AWS:AWSComponent"
```

- Per API i dettagli, vedere [GetInventorySchema](#) in AWS CLI Command Reference.

## get-inventory

Il seguente esempio di codice mostra come utilizzare `get-inventory`.

## AWS CLI

Per visualizzare il tuo inventario

Questo esempio ottiene i metadati personalizzati per il tuo inventario.

Comando:

```
aws ssm get-inventory
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, consulta [GetInventory AWS CLI Command Reference](#).



## get-maintenance-window-execution-task-invocation

Il seguente esempio di codice mostra come utilizzare `get-maintenance-window-execution-task-invocation`.

### AWS CLI

Per ottenere informazioni su una finestra di manutenzione, invocazione di un'operazione

L'`get-maintenance-window-execution-task-invocation` seguente elenca le informazioni sulla chiamata dell'attività specificata che fa parte dell'esecuzione della finestra di manutenzione specificata.

```
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"
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, vedere [View Information About Tasks and Task Executions \(AWS CLI\)](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [GetMaintenanceWindowExecutionTaskInvocation](#) in AWS CLI Command Reference.

## get-maintenance-window-execution-task

Il seguente esempio di codice mostra come utilizzare `get-maintenance-window-execution-task`.

### AWS CLI

Per ottenere informazioni sull'esecuzione di un'operazione in una finestra di manutenzione

L'`get-maintenance-window-execution-task`esempio seguente elenca le informazioni su un'attività che fa parte dell'esecuzione della finestra di manutenzione specificata.

```
aws ssm get-maintenance-window-execution-task \
  --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE" \
  --task-id "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
```

### Output:

```
{
  "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"
}

```

Per ulteriori informazioni, vedere [Visualizza informazioni sulle attività e sulle esecuzioni delle attività \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [GetMaintenanceWindowExecutionTask](#) in AWS CLI Command Reference.

## get-maintenance-window-execution

Il seguente esempio di codice mostra come utilizzare `get-maintenance-window-execution`.

### AWS CLI

Per ottenere informazioni sull'esecuzione di un'operazione in una finestra di manutenzione

L'esempio seguente elenca le informazioni su un'attività eseguita come parte dell'esecuzione della finestra di manutenzione specificata.

```
aws ssm get-maintenance-window-execution \  
  --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

Output:

```
{  
  "Status": "SUCCESS",  
  "TaskIds": [  
    "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"  
  ],  
  "StartTime": 1487692834.595,  
  "EndTime": 1487692835.051,  
  "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",  
}
```

Per ulteriori informazioni, vedere [View Information About Tasks and Task Executions \(AWS CLI\)](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [GetMaintenanceWindowExecution](#) in AWS CLI Command Reference.

## get-maintenance-window-task

Il seguente esempio di codice mostra come utilizzare `get-maintenance-window-task`.

AWS CLI

Per ottenere informazioni su un'operazione relativa alla finestra di manutenzione

L'`get-maintenance-window-task`esempio seguente recupera i dettagli sull'attività specificata nella finestra di manutenzione.

```
aws ssm get-maintenance-window-task \  
  --window-id mw-0c5ed765acEXAMPLE \  
  --window-task-id 0e842a8d-2d44-4886-bb62-af8dcEXAMPLE
```

Output:

```
{  
  "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"
}

```

Per ulteriori informazioni, vedere [Visualizza informazioni sulla manutenzione di Windows \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [GetMaintenanceWindowTask](#) in AWS CLI Command Reference.

## get-maintenance-window

Il seguente esempio di codice mostra come utilizzare `get-maintenance-window`.

## AWS CLI

Per ottenere informazioni su una finestra di manutenzione

L'`get-maintenance-window` seguente recupera i dettagli sulla finestra di manutenzione specificata.

```
aws ssm get-maintenance-window \  
  --window-id "mw-03eb9db428EXAMPLE"
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, vedere [Visualizza informazioni sulle finestre di manutenzione \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [GetMaintenanceWindow](#) in AWS CLI Command Reference.

## get-ops-item

Il seguente esempio di codice mostra come utilizzare `get-ops-item`.

### AWS CLI

Per visualizzare informazioni su un OpsItem

L'`get-ops-item` seguente visualizza i dettagli relativi a quanto specificato OpsItem.

```
aws ssm get-ops-item \  
  --ops-item-id oi-0b725EXAMPLE
```

## Output:

```

{
  "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"
    }
}
```

Per ulteriori informazioni, vedere [Working with OpsItems](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [GetOpsItem](#) in AWS CLI Command Reference.

## get-ops-summary

Il seguente esempio di codice mostra come utilizzare `get-ops-summary`.

### AWS CLI

Per visualizzare un riepilogo di tutti OpsItems

L'`get-ops-summary` esempio seguente mostra un riepilogo di tutti i dati OpsItems presenti nel tuo AWS account.

```
aws ssm get-ops-summary
```

Output:

```
{
  "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 SSM0psItems-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"
    }
  ]
}

```

Per ulteriori informazioni, vedere [Working with OpsItems](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [GetOpsSummary](#) in AWS CLI Command Reference.

## get-parameter-history

Il seguente esempio di codice mostra come utilizzare `get-parameter-history`.

### AWS CLI

Per ottenere una cronologia dei valori per un parametro

L'esempio seguente elenca la cronologia delle modifiche per il parametro specificato, incluso il relativo valore.

```
aws ssm get-parameter-history \
  --name "MyStringParameter"
```

Output:

```
{
```

```
"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": []  
  }  
]  
}
```

Per ulteriori informazioni, vedere [Utilizzo delle versioni dei parametri](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [GetParameterHistory](#) in AWS CLI Command Reference.

## get-parameter

Il seguente esempio di codice mostra come utilizzare `get-parameter`.

### AWS CLI

Esempio 1: Per visualizzare il valore di un parametro

L'`get-parameter` esempio seguente elenca il valore per il singolo parametro specificato.

```
aws ssm get-parameter \  
  --name "MyStringParameter"
```

Output:

```
{  
  "Parameter": {  
    "Name": "MyStringParameter",  
    "Type": "String",  
    "Value": "Veni",  
    "Version": 1,  
    "LastModifiedDate": 1530018761.888,  
    "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"  
    "DataType": "text"  
  }  
}
```

Per ulteriori informazioni, vedere [Working with Parameter Store](#) nella AWS Systems Manager User Guide.

Esempio 2: per decrittografare il valore di un parametro `SecureString`

L'`get-parameter` esempio seguente decrittografa il valore del parametro specificato.

`SecureString`

```
aws ssm get-parameter \  
  --name "MySecureStringParameter" \  
  --with-decryption
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, vedere [Working with Parameter Store](#) nella AWS Systems Manager User Guide.

Esempio 3: Per visualizzare il valore di un parametro utilizzando le etichette

L'get-parameter esempio seguente elenca il valore per il singolo parametro specificato con un'etichetta specificata.

```
aws ssm get-parameter \
  --name "MyParameter:label"
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, vedere [Utilizzo delle etichette dei parametri](#) nella Guida per l'utente di AWS Systems Manager.

Esempio 4: Per visualizzare il valore di un parametro utilizzando le versioni

L'get-parameteresempio seguente elenca il valore per la versione a parametro singolo specificata.

```
aws ssm get-parameter \  
  --name "MyParameter:2"
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [Utilizzo delle etichette dei parametri](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [GetParameter](#) in AWS CLI Command Reference.

## get-parameters-by-path

Il seguente esempio di codice mostra come utilizzareget-parameters-by-path.

AWS CLI

Per elencare i parametri in un percorso specifico

L'get-parameters-by-pathesempio seguente elenca i parametri all'interno della gerarchia specificata.

```
aws ssm get-parameters-by-path \  
  --path "/site/newyork/department/"
```

**Output:**

```
{
  "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"
    },
    ...
  ]
}
```

Per ulteriori informazioni, vedere [Utilizzo delle gerarchie di parametri](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [GetParametersByPath](#) in AWS CLI Command Reference.

**get-parameters**

Il seguente esempio di codice mostra come utilizzare `get-parameters`.

**AWS CLI**

Esempio 1: Per elencare i valori di un parametro

L'`get-parameters` seguente elenca i valori per i tre parametri specificati.

```
aws ssm get-parameters \
  --names "MyStringParameter" "MyStringListParameter" "MyInvalidParameterName"
```

## Output:

```
{
  "Parameters": [
    {
      "Name": "MyStringListParameter",
      "Type": "StringList",
      "Value": "alpha,beta,gamma",
      "Version": 1,
      "LastModifiedDate": 1582154764.222,
      "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringListParameter"
    },
    {
      "Name": "MyStringParameter",
      "Type": "String",
      "Value": "Vici",
      "Version": 3,
      "LastModifiedDate": 1582156117.545,
      "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"
    }
  ],
  "InvalidParameters": [
    "MyInvalidParameterName"
  ]
}
```

Per ulteriori informazioni, vedere [Working with Parameter Store](#) nella AWS Systems Manager User Guide.

Esempio 2: Per elencare i nomi e i valori di più parametri utilizzando l'opzione ``--query``

L'get-parametersesempio seguente elenca i nomi e i valori dei parametri specificati.

```
aws ssm get-parameters \
  --names MyStringParameter MyStringListParameter \
  --query "Parameters[*].{Name:Name,Value:Value}"
```

## Output:

```
[
```



```
{
  "Name": "MyStringListParameter",
  "Value": "alpha,beta,gamma"
},
{
  "Name": "MyStringParameter",
  "Value": "Vidi"
}
]
```

Per ulteriori informazioni, vedere [Working with Parameter Store](#) nella AWS Systems Manager User Guide.

Esempio 3: Per visualizzare il valore di un parametro utilizzando le etichette

L'get-parameteresempio seguente elenca il valore per il singolo parametro specificato con un'etichetta specificata.

```
aws ssm get-parameter \
  --name "MyParameter:label"
```

Output:

```
{
  "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": []
  }
```

Per ulteriori informazioni, vedere [Utilizzo delle etichette dei parametri](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [GetParameters](#) in AWS CLI Command Reference.

## get-patch-baseline-for-patch-group

Il seguente esempio di codice mostra come utilizzare `get-patch-baseline-for-patch-group`.

### AWS CLI

Per visualizzare la linea di base delle patch per un gruppo di patch

L'`get-patch-baseline-for-patch-group` esempio seguente recupera i dettagli sulla linea di base delle patch per il gruppo di patch specificato.

```
aws ssm get-patch-baseline-for-patch-group \
  --patch-group "DEV"
```

Output:

```
{
  "PatchGroup": "DEV",
  "BaselineId": "pb-0123456789abcdef0",
  "OperatingSystem": "WINDOWS"
}
```

Per ulteriori informazioni, vedere [Create a Patch Group < https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>](https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html) e [Aggiungere un gruppo di patch a una Patch Baseline nella Systems Manager AWS User Guide](#).

- Per i API dettagli, vedere in Command Reference. [GetPatchBaselineForPatchGroup](#) AWS CLI

## get-patch-baseline

Il seguente esempio di codice mostra come utilizzare `get-patch-baseline`.

## AWS CLI

Per visualizzare una linea di base della patch

L'get-patch-baselinesesempio seguente recupera i dettagli per la baseline della patch specificata.

```
aws ssm get-patch-baseline \  
--baseline-id "pb-0123456789abcdef0"
```

Output:

```
{  
  "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": []
}
```

Per ulteriori informazioni, vedere [Informazioni sulle patch di base nella Guida per l'utente di AWS Systems Manager](#).

- Per API i dettagli, vedere [GetPatchBaseline](#) in AWS CLI Command Reference.

## get-service-setting

Il seguente esempio di codice mostra come utilizzare `get-service-setting`.

### AWS CLI

Per recuperare l'impostazione del servizio per il throughput di Parameter Store

Di seguito `get-service-setting` Questo esempio recupera l'impostazione corrente del servizio per la velocità effettiva di Parameter Store nella regione specificata.

```
aws ssm get-service-setting \
  --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-
  store/high-throughput-enabled
```

Output:

```
{
  "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"
  }
}
```

```
}
```

Per ulteriori informazioni, vedere [Increasing Parameter Store Throughput](#) nella AWS Systems Manager Users Guide.

- Per API i dettagli, vedere [GetServiceSetting](#) in AWS CLI Command Reference.

## label-parameter-version

Il seguente esempio di codice mostra come utilizzare `label-parameter-version`.

### AWS CLI

Esempio 1: aggiungere un'etichetta alla versione più recente di un parametro

L'`label-parameter-version`esempio seguente aggiunge un'etichetta alla versione più recente del parametro specificato.

```
aws ssm label-parameter-version \  
  --name "MyStringParameter" \  
  --labels "ProductionReady"
```

Output:

```
{  
  "InvalidLabels": [],  
  "ParameterVersion": 3  
}
```

Per ulteriori informazioni, vedere [Utilizzo delle etichette dei parametri](#) nella Guida per l'utente di AWS Systems Manager.

Esempio 2: aggiungere un'etichetta a una versione specifica di un parametro

L'`label-parameter-version`esempio seguente aggiunge un'etichetta alla versione specificata di un parametro.

```
aws ssm label-parameter-version \  
  --name "MyStringParameter" \  
  --labels "ProductionReady" \  
  --parameter-version "2" --labels "DevelopmentReady"
```

Per ulteriori informazioni, vedere [Utilizzo delle etichette dei parametri](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [LabelParameterVersion](#) in AWS CLI Command Reference.

## list-association-versions

Il seguente esempio di codice mostra come utilizzare `list-association-versions`.

### AWS CLI

Per elencare tutte le versioni di un'associazione per un ID di associazione specifico

L'`list-association-versions` seguente elenca tutte le versioni delle associazioni specificate.

```
aws ssm list-association-versions \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

### Output:

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, vedere [Utilizzo delle associazioni in Systems Manager](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [ListAssociationVersions](#) in AWS CLI Command Reference.

## list-associations

Il seguente esempio di codice mostra come utilizzare `list-associations`.

### AWS CLI

Esempio 1: per elencare le associazioni per un'istanza specifica

Il seguente esempio di associazioni di elenchi elenca tutte le associazioni con `AssociationName`, `U.pdateSSMAgent`

```

aws ssm list-associations /
  --association-filter-list "key=AssociationName,value=UpdateSSMAgent"

```

Output:

```

{
  "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"
  }
]
}

```

Per ulteriori informazioni, vedere [Utilizzo delle associazioni in Systems Manager](#) nella Guida per l'utente di Systems Manager.

Esempio 2: Per elencare le associazioni per un documento specifico

Il seguente esempio di associazioni di elenchi elenca tutte le associazioni per il documento specificato.

```

aws ssm list-associations /
  --association-filter-list "key=Name,value=AWS-UpdateSSMAgent"

```

Output:

```

{
  "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
      }
    }
  }
]
}

```

Per ulteriori informazioni, vedere [Utilizzo delle associazioni in Systems Manager](#) nella Guida per l'utente di Systems Manager.

- Per API i dettagli, vedere [ListAssociations](#) in AWS CLI Command Reference.

## list-command-invocations

Il seguente esempio di codice mostra come utilizzare `list-command-invocations`.

## AWS CLI

Per elencare le invocazioni di un comando specifico

L'`list-command-invocations` seguente elenca tutte le invocazioni di un comando.

```
aws ssm list-command-invocations \  
  --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \  
  --details
```

Output:

```
{  
  "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": "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": "",  
          "StandardErrorUrl": "",  
          "OutputS3Region": "us-east-2",  
          "OutputS3BucketName": "",
```

```

        "OutputS3KeyPrefix": ""
    }
],
"ServiceRole": "",
"NotificationConfig": {
    "NotificationArn": "",
    "NotificationEvents": [],
    "NotificationType": ""
},
"CloudWatchOutputConfig": {
    "CloudWatchLogGroupName": "",
    "CloudWatchOutputEnabled": false
}
},
{
    "CommandId": "ef7fdfd8-9b57-4151-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
  }
}
]
}

```

Per ulteriori informazioni, vedere [Understanding Command Statuses](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [ListCommandInvocations](#) in AWS CLI Command Reference.

## list-commands

Il seguente esempio di codice mostra come utilizzare `list-commands`.

### AWS CLI

Esempio 1: Per ottenere lo stato di un comando specifico

L'`list-commands` esempio seguente recupera e visualizza lo stato del comando specificato.

```

aws ssm list-commands \
  --command-id "0831e1a8-a1ac-4257-a1fd-c831bEXAMPLE"

```

Esempio 2: Per ottenere lo stato dei comandi richiesti dopo una data specifica

L'`list-commands` seguente recupera i dettagli dei comandi richiesti dopo la data specificata.

```
aws ssm list-commands \  
  --filter "key=InvokedAfter,value=2020-02-01T00:00:00Z"
```

Esempio 3: Per elencare tutti i comandi richiesti in un account AWS

L'`list-commands` seguente elenca tutti i comandi richiesti dagli utenti nell' AWS account e nella regione correnti.

```
aws ssm list-commands
```

Output:

```
{  
  "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
    }
}
]
}

```

Per ulteriori informazioni, vedere [Running Commands Using Systems Manager Run Command](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [ListCommands](#) in AWS CLI Command Reference.

## list-compliance-items

Il seguente esempio di codice mostra come utilizzare `list-compliance-items`.



## AWS CLI

Per elencare gli elementi di conformità per un'istanza specifica

Questo esempio elenca tutti gli elementi di conformità per l'istanza specificata.

Comando:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-  
types "ManagedInstance"
```

Output:

```
{  
  "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--"
}

```

Per elencare gli elementi di conformità per un'istanza e un ID di associazione specifici

Questo esempio elenca tutti gli elementi di conformità per l'ID di istanza e associazione specificati.

Comando:

```

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"

```

Per elencare gli elementi di conformità per un'istanza dopo una data e un'ora specifiche

Questo esempio elenca tutti gli elementi di conformità per un'istanza dopo la data e l'ora specificate.

Comando:

```

aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ExecutionTime,Values=2019-02-18T16:00:00Z,Type=GREATER_THAN"

```

- Per API i dettagli, vedere [ListComplianceItems](#) in AWS CLI Command Reference.

## list-compliance-summaries

Il seguente esempio di codice mostra come utilizzare `list-compliance-summaries`.

AWS CLI

Per elencare i riepiloghi di conformità per tutti i tipi di conformità

Questo esempio elenca i riepiloghi di conformità per tutti i tipi di conformità presenti nel tuo account.

Comando:

```
aws ssm list-compliance-summaries
```

Output:

```
{
  "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=="
}

```

Per elencare i riepiloghi di conformità per un tipo di conformità specifico

Questo esempio elenca il riepilogo della conformità per il tipo di conformità Patch.

Comando:

```
aws ssm list-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- Per API i dettagli, vedere [ListComplianceSummaries](#) in AWS CLI Command Reference.

## list-document-metadata-history

Il seguente esempio di codice mostra come utilizzare `list-document-metadata-history`.

AWS CLI

Esempio: per visualizzare la cronologia e lo stato delle approvazioni di un modello di modifica

L'`list-document-metadata-history` esempio seguente restituisce la cronologia delle approvazioni per il modello di modifica di Change Manager specificato.

```
aws ssm list-document-metadata-history \
  --name MyChangeManageTemplate \
  --metadata DocumentReviews
```

Output:

```
{
  "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"
      }
    ]
  }
}
```

Per ulteriori informazioni, vedere [Revisione e approvazione o rifiuto dei modelli di modifica](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [ListDocumentMetadataHistory](#) in AWS CLI Command Reference.

## list-document-versions

Il seguente esempio di codice mostra come utilizzare `list-document-versions`.

## AWS CLI

Per elencare le versioni dei documenti

L'`list-document-versions` seguente elenca tutte le versioni di un documento Systems Manager.

```
aws ssm list-document-versions \  
  --name "Example"
```

Output:

```
{  
  "DocumentVersions": [  
    {  
      "Name": "Example",  
      "DocumentVersion": "1",  
      "CreateDate": 1583257938.266,  
      "IsDefaultVersion": true,  
      "DocumentFormat": "YAML",  
      "Status": "Active"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Invio di comandi che utilizzano il parametro della versione del documento](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [ListDocumentVersions](#) in AWS CLI Command Reference.

## list-documents

Il seguente esempio di codice mostra come utilizzare `list-documents`.

### AWS CLI

Esempio 1: per elencare i documenti

L'`list-documents` seguente elenca i documenti di proprietà dell'account richiedente contrassegnati con il tag personalizzato.

```
aws ssm list-documents \  
  --tag-key "tag-key"
```

```
--filters Key=Owner,Values=Self Key=tag:DocUse,Values=Testing
```

Output:

```
{
  "DocumentIdentifiers": [
    {
      "Name": "Example",
      "Owner": "29884EXAMPLE",
      "PlatformTypes": [
        "Windows",
        "Linux"
      ],
      "DocumentVersion": "1",
      "DocumentType": "Automation",
      "SchemaVersion": "0.3",
      "DocumentFormat": "YAML",
      "Tags": [
        {
          "Key": "DocUse",
          "Value": "Testing"
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni, vedere [AWS i documenti di Systems Manager](#) nella Guida per l'utente di AWS Systems Manager.

Esempio 2: per elencare i documenti condivisi

L'`list-documents` seguente elenca i documenti condivisi, inclusi i documenti privati condivisi non di proprietà di AWS.

```
aws ssm list-documents \  
--filters Key=Name,Values=sharedDocNamePrefix Key=Owner,Values=Private
```

Output:

```
{
  "DocumentIdentifiers": [
```

```
{
  "Name": "Example",
  "Owner": "12345EXAMPLE",
  "PlatformTypes": [
    "Windows",
    "Linux"
  ],
  "DocumentVersion": "1",
  "DocumentType": "Command",
  "SchemaVersion": "0.3",
  "DocumentFormat": "YAML",
  "Tags": []
}
]
```

Per ulteriori informazioni, vedere [AWS i documenti di Systems Manager](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [ListDocuments](#) in AWS CLI Command Reference.

## list-inventory-entries

Il seguente esempio di codice mostra come utilizzare `list-inventory-entries`.

### AWS CLI

Esempio 1: per visualizzare le voci relative al tipo di inventario specifiche per un'istanza

L'`list-inventory-entries` esempio seguente elenca le voci di inventario per il tipo di AWS inventario: Application su un'istanza specifica.

```
aws ssm list-inventory-entries \
  --instance-id "i-1234567890abcdef0" \
  --type-name "AWS:Application"
```

Output:

```
{
  "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"
  }
]
}

```

Esempio 2: per visualizzare le voci di inventario personalizzate assegnate a un'istanza

L'`list-inventory-entries` seguente elenca una voce di inventario personalizzata assegnata a un'istanza.

```

aws ssm list-inventory-entries \
  --instance-id "i-1234567890abcdef0" \
  --type-name "Custom:RackInfo"

```

Output:

```

{
  "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"
    }
  ]
}

```

- Per API i dettagli, vedere [ListInventoryEntries](#) in AWS CLI Command Reference.

## list-ops-item-related-items

Il seguente esempio di codice mostra come utilizzare `list-ops-item-related-items`.

### AWS CLI

Per elencare le risorse relative agli elementi correlati di un OpsItem

L'esempio seguente elenca le risorse relative agli elementi correlati di un OpsItem

```
aws ssm list-ops-item-related-items \  
  --ops-item-id "oi-f99f2EXAMPLE"
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Utilizzo degli incidenti di Incident Manager OpsCenter nella Guida per l'utente di AWS Systems Manager](#).

- Per API i dettagli, vedere [ListOpsItemRelatedItems](#) in AWS CLI Command Reference.

## list-resource-compliance-summaries

Il seguente esempio di codice mostra come utilizzare `list-resource-compliance-summaries`.

### AWS CLI

Per elencare i conteggi riepilogativi della conformità a livello di risorsa

Questo esempio elenca i conteggi riepilogativi della conformità a livello di risorsa.

Comando:

```
aws ssm list-resource-compliance-summaries
```

Output:

```
{
  "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--"
}
```

Per elencare i riepiloghi di conformità a livello di risorsa per un tipo di conformità specifico

Questo esempio elenca i riepiloghi di conformità a livello di risorsa per il tipo di conformità Patch.

Comando:

```
aws ssm list-resource-compliance-summaries --  
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- Per i API dettagli, vedere in Command Reference. [ListResourceComplianceSummaries](#) AWS CLI

## list-resource-data-sync

Il seguente esempio di codice mostra come utilizzare `list-resource-data-sync`.

AWS CLI

Per elencare le configurazioni di sincronizzazione dei dati delle risorse

Questo esempio recupera informazioni sulle configurazioni di sincronizzazione dei dati delle risorse.

```
aws ssm list-resource-data-sync
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

```
]
}
```

- Per API i dettagli, vedere [ListResourceDataSync](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag applicati a una patch di base

L'`list-tags-for-resource` esempio seguente elenca i tag per una patch baseline.

```
aws ssm list-tags-for-resource \
  --resource-type "PatchBaseline" \
  --resource-id "pb-0123456789abcdef0"
```

Output:

```
{
  "TagList": [
    {
      "Key": "Environment",
      "Value": "Production"
    },
    {
      "Key": "Region",
      "Value": "EMEA"
    }
  ]
}
```

Per ulteriori informazioni, vedete [Tagging AWS Resources](#) nella Guida AWS generale.

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## modify-document-permission

Il seguente esempio di codice mostra come utilizzare `modify-document-permission`.

## AWS CLI

Per modificare le autorizzazioni dei documenti

L'esempio seguente condivide pubblicamente un documento Systems Manager.

```
aws ssm modify-document-permission \  
  --name "Example" \  
  --permission-type "Share" \  
  --account-ids-to-add "All"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Share a Systems Manager Document](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [ModifyDocumentPermission](#) in AWS CLI Command Reference.

## put-compliance-items

Il seguente esempio di codice mostra come utilizzare `put-compliance-items`.

### AWS CLI

Per registrare un tipo di conformità e i dettagli di conformità in un'istanza designata

Questo esempio registra il tipo di `Custom:AVCheck` conformità nell'istanza gestita specificata. Se il comando va a buon fine, non viene generato output.

Comando:

```
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"
```

- Per API i dettagli, vedere [PutComplianceItems](#) in AWS CLI Command Reference.

## put-inventory

Il seguente esempio di codice mostra come utilizzare `put-inventory`.

## AWS CLI

Per assegnare i metadati dei clienti a un'istanza

Questo esempio assegna le informazioni sulla posizione del rack a un'istanza. Se il comando va a buon fine, non viene generato output.

Comando (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"}]} ]'
```

Comando (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']]"
```

- Per API i dettagli, vedere [PutInventory](#) in AWS CLI Command Reference.

## put-parameter

Il seguente esempio di codice mostra come utilizzare `put-parameter`.

### AWS CLI

Esempio 1: modificare il valore di un parametro

L'`put-parameter` esempio seguente modifica il valore del parametro specificato.

```
aws ssm put-parameter \
  --name "MyStringParameter" \
  --type "String" \
  --value "Vici" \
  --overwrite
```

Output:

```
{
```



```
"Version": 2,  
"Tier": "Standard"  
}
```

Per ulteriori informazioni, vedere [Create a Systems Manager parameter \(AWS CLI\)](https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html), 'Managing parametric tiers < <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>>' e [Working with parameters policy nella](#) Systems Manager User Guide.AWS

Esempio 2: per creare un parametro avanzato

L'put-parameteresempio seguente crea un parametro avanzato.

```
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
```

Output:

```
{  
  "Version": 1,  
  "Tier": "Advanced"  
}
```

Per ulteriori informazioni, vedere [Create a Systems Manager parameter \(AWS CLI\)](https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html), 'Managing parametric tiers < <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>>' e [Working with parameters policy nella](#) Systems Manager User Guide.AWS

Esempio 3: convertire un parametro standard in un parametro avanzato

L'put-parameteresempio seguente converte un parametro standard esistente in un parametro avanzato.

```
aws ssm put-parameter \  

```

```
--name "MyConvertedParameter" \
--value "abc123" \
--type "String" \
--tier Advanced \
--overwrite
```

Output:

```
{
  "Version": 2,
  "Tier": "Advanced"
}
```

Per ulteriori informazioni, vedere [Create a Systems Manager parameter \(AWS CLI\)](https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html), 'Managing parametric tiers < <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>>' e [Working with parameters policy nella Systems Manager User Guide.AWS](#)

Esempio 4: creare un parametro con una policy allegata

L'put-parameteresempio seguente crea un parametro avanzato con una politica dei parametri allegata.

```
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"}}]"
```

Output:

```
{
  "Version": 1,
  "Tier": "Advanced"
}
```

Per ulteriori informazioni, vedere [Create a Systems Manager parameter \(AWS CLI\)](https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html), 'Managing parameters tiers < <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>>' e [Working with parameters policy nella](#) Systems Manager User Guide.AWS

Esempio 5: aggiungere una policy a un parametro esistente

L'put-parametereseempio seguente allega una politica a un parametro avanzato esistente.

```
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
```

Output:

```
{
  "Version": 2,
  "Tier": "Advanced"
}
```

Per ulteriori informazioni, vedere [Create a Systems Manager parameter \(AWS CLI\)](https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html), 'Managing parameters tiers < <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>>' e [Working with parameters policy nella](#) Systems Manager User Guide.AWS

- Per API i dettagli, vedere in Command Reference. [PutParameter](#)AWS CLI

## register-default-patch-baseline

Il seguente esempio di codice mostra come utilizzare register-default-patch-baseline.

AWS CLI

Per impostare la linea di base della patch predefinita

L'`register-default-patch-baseline` seguente registra la patch baseline personalizzata specificata come baseline di patch predefinita per il tipo di sistema operativo che supporta.

```
aws ssm register-default-patch-baseline \  
  --baseline-id "pb-abc123cf9bEXAMPLE"
```

Output:

```
{  
  "BaselineId": "pb-abc123cf9bEXAMPLE"  
}
```

L'`register-default-patch-baseline` seguente registra la linea di base delle patch predefinita fornita da per AWS CentOS come linea di base delle patch predefinita.

```
aws ssm register-default-patch-baseline \  
  --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/  
pb-0574b43a65ea646ed"
```

Output:

```
{  
  "BaselineId": "pb-abc123cf9bEXAMPLE"  
}
```

Per ulteriori informazioni, vedere [Informazioni sulle patch di base predefinite e personalizzate nella Guida per l'utente](#) di AWS Systems Manager.

- Per API i dettagli, vedere [RegisterDefaultPatchBaseline](#) in AWS CLI Command Reference.

## **register-patch-baseline-for-patch-group**

Il seguente esempio di codice mostra come utilizzare `register-patch-baseline-for-patch-group`.

AWS CLI

Per registrare una linea di base di patch per un gruppo di patch

L'`register-patch-baseline-for-patch-group`esempio seguente registra una linea di base di patch per un gruppo di patch.

```
aws ssm register-patch-baseline-for-patch-group \  
  --baseline-id "pb-045f10b4f382baeda" \  
  --patch-group "Production"
```

Output:

```
{  
  "BaselineId": "pb-045f10b4f382baeda",  
  "PatchGroup": "Production"  
}
```

Per ulteriori informazioni, vedere [Create a Patch Group < https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>](https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html) e [Aggiungere un gruppo di patch a una Patch Baseline nella Systems Manager AWS User Guide](#).

- Per i API dettagli, vedere in Command Reference. [RegisterPatchBaselineForPatchGroup](#)AWS CLI

## **register-target-with-maintenance-window**

Il seguente esempio di codice mostra come utilizzare `register-target-with-maintenance-window`.

AWS CLI

Esempio 1: registrare un singolo obiettivo con una finestra di manutenzione

L'`register-target-with-maintenance-window`esempio seguente registra un'istanza con una finestra di manutenzione.

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-ab12cd34ef56gh78" \  
  --target "Key=InstanceIds,Values=i-0000293ffd8c57862" \  
  --owner-information "Single instance" \  
  --resource-type "INSTANCE"
```

Output:

```
{
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Esempio 2: per registrare più destinazioni con una finestra di manutenzione utilizzando l'istanza IDs

L'`register-target-with-maintenance-window` seguente registra due istanze con una finestra di manutenzione specificando la relativa istanza. 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"
```

Output:

```
{
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Esempio 3: registrare gli obiettivi in una finestra di manutenzione utilizzando i tag delle risorse

L'`register-target-with-maintenance-window` seguente registra le istanze con una finestra di manutenzione specificando i tag di risorsa che sono stati applicati alle istanze.

```
aws ssm register-target-with-maintenance-window \
  --window-id "mw-06cf17cbefcb4bf4f" \
  --targets "Key=tag:Environment,Values=Prod" "Key=Role,Values=Web" \
  --owner-information "Production Web Servers" \
  --resource-type "INSTANCE"
```

Output:

```
{
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

Esempio 4: registrare gli obiettivi utilizzando un gruppo di chiavi di tag

L'esempio seguente registra le istanze a cui sono tutte assegnate una o più chiavi di tag, indipendentemente dai relativi valori di chiave.

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-0c50858d01EXAMPLE" \  
  --resource-type "INSTANCE" \  
  --target "Key=tag-key, Values=Name, Instance-Type, CostCenter"
```

Output:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

Esempio 5: registrare gli obiettivi utilizzando il nome di un gruppo di risorse

L'esempio seguente registra un gruppo di risorse specificato, indipendentemente dal tipo di risorse che contiene.

```
aws ssm register-target-with-maintenance-window \  
  --window-id "mw-0c50858d01EXAMPLE" \  
  --resource-type "RESOURCE_GROUP" \  
  --target "Key=resource-groups:Name, Values=MyResourceGroup"
```

Output:

```
{  
  "WindowTargetId": "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"  
}
```

Per ulteriori informazioni, vedere [Registrazione un'istanza di Target con la finestra di manutenzione \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [RegisterTargetWithMaintenanceWindow](#) in AWS CLI Command Reference.

## register-task-with-maintenance-window

Il seguente esempio di codice mostra come utilizzare `register-task-with-maintenance-window`.

## AWS CLI

Esempio 1: registrare un'attività di automazione con una finestra di manutenzione

L'`register-task-with-maintenance-window` seguente registra un'attività di automazione con una finestra di manutenzione destinata a un'istanza.

```
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"
```

Output:

```
{
  "WindowTaskId": "11144444-5555-6666-7777-88888888"
}
```

Per ulteriori informazioni, vedere [Registrazione un'attività con la finestra di manutenzione \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

Esempio 2: registrare un'attività Lambda con una finestra di manutenzione

L'`register-task-with-maintenance-window` seguente registra un'attività Lambda con una finestra di manutenzione destinata a un'istanza.

```
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"
```

Output:

```
{
  "WindowTaskId": "22244444-5555-6666-7777-88888888"
}
```

Per ulteriori informazioni, vedere [Registrazione un'attività con la finestra di manutenzione \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

Esempio 3: registrare un'attività di esecuzione del comando in una finestra di manutenzione

L'operazione `register-task-with-maintenance-window` seguente registra un'attività Run Command con una finestra di manutenzione destinata a un'istanza.

```
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
```

Output:

```
{
  "WindowTaskId": "33344444-5555-6666-7777-88888888"
}
```

Per ulteriori informazioni, vedere [Registrazione un'attività con la finestra di manutenzione \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

Esempio 4: registrare un'attività Step Functions con una finestra di manutenzione

L'`register-task-with-maintenance-window` seguente registra un'attività Step Functions con una finestra di manutenzione destinata a un'istanza.

```
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"
```

Output:

```
{
  "WindowTaskId": "444444444-5555-6666-7777-88888888"
}
```

Per ulteriori informazioni, vedere [Registrazione un'attività con la finestra di manutenzione \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

Esempio 5: per registrare un'attività utilizzando un ID di destinazione di Windows di manutenzione

L'`register-task-with-maintenance-window` seguente registra un'attività utilizzando un ID di destinazione della finestra di manutenzione. L'ID di destinazione della finestra di manutenzione era nell'output del `aws ssm register-target-with-maintenance-window` comando. È inoltre possibile recuperarlo dall'output del `aws ssm describe-maintenance-window-targets` comando.

```
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

```

Output:

```

{
  "WindowTaskId": "33344444-5555-6666-7777-88888888"
}

```

Per ulteriori informazioni, vedere [Registrazione un'attività con la finestra di manutenzione \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [RegisterTaskWithMaintenanceWindow](#) in AWS CLI Command Reference.

## remove-tags-from-resource

Il seguente esempio di codice mostra come utilizzare `remove-tags-from-resource`.

AWS CLI

Per rimuovere un tag dalla linea di base di una patch

L'`remove-tags-from-resource` esempio seguente rimuove i tag da una patch baseline.

```

aws ssm remove-tags-from-resource \
  --resource-type "PatchBaseline" \
  --resource-id "pb-0123456789abcdef0" \
  --tag-keys "Region"

```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedete [Tagging AWS Resources](#) nella Guida AWS generale.

- Per API i dettagli, vedere [RemoveTagsFromResource](#) in AWS CLI Command Reference.

## reset-service-setting

Il seguente esempio di codice mostra come utilizzare `reset-service-setting`.

### AWS CLI

Per ripristinare l'impostazione del servizio per il throughput di Parameter Store

L'`reset-service-setting` seguente reimposta l'impostazione del servizio per la velocità effettiva di Parameter Store nella regione specificata in modo che non utilizzi più una velocità effettiva maggiore.

```
aws ssm reset-service-setting \
  --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-
  store/high-throughput-enabled
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, vedere [Increasing Parameter Store Throughput](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [ResetServiceSetting](#) in AWS CLI Command Reference.

## resume-session

Il seguente esempio di codice mostra come utilizzare `resume-session`.

### AWS CLI

Per riprendere una sessione di Session Manager

Questo `resume-session` esempio riprende una sessione di Session Manager con un'istanza dopo che questa è stata disconnessa. Si noti che questo comando interattivo richiede l'installazione del plug-in Session Manager sul computer client che effettua la chiamata.

```
aws ssm resume-session \  
  --session-id Mary-Major-07a16060613c408b5
```

Output:

```
{  
  "SessionId": "Mary-Major-07a16060613c408b5",  
  "TokenValue":  
    "AAEAAVbTGsa0nyvcUoNGqifbv5r/8l9xuQ1jCuY8qVcv0noBAAAAAFxtd3jIXAFUUXGTJ7zF/  
AWJpWdvi0lF5p3d1AgrqVIV06IEXhkHLz0/1gXKRKEME71E6TL0p1LDJAMZ  
+kREejkZu4c5AxMkrQjMF+gtHP1bYJKTwtHQd1wjulPLex08SH17g5R/  
wekrj6WsDUpnEegFBfGftpAIz2GXQVfTJXKfkc5qepQ11C11D0IT2doz0qXgHwfQHfAKLErM5dWDZqKwyT1Z3iw7unQd  
+ihfGa6MEJJ97Jmat/a2TspEn0jNn9Mvu5iwXIW2yCvWZrGUj+/  
QI5Xr7s1XJBEEnSKR54o4fN0GV9RWl0RZsZm1m1ki0JJtiwwgZ",  
  "StreamUrl": "wss://ssmmessages.us-east-2.amazonaws.com/v1/data-channel/Mary-  
Major-07a16060613c408b5?role=publish_subscribe"  
}
```

Per ulteriori informazioni, vedere [Installare il plug-in Session Manager AWS CLI nella Guida per l'utente di AWS Systems Manager](#).

- Per API i dettagli, vedere [ResumeSession](#) in AWS CLI Command Reference.

## send-automation-signal

Il seguente esempio di codice mostra come utilizzare `send-automation-signal`.

### AWS CLI

Inviare un segnale a un'esecuzione di automazione

L'`send-automation-signal` esempio seguente invia un segnale di approvazione a un'esecuzione di automazione.

```
aws ssm send-automation-signal \  
  --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \  
  --signal-type "Approve"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Esecuzione di un flusso di lavoro di automazione con approvatori nella Guida](#) per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [SendAutomationSignal](#) in AWS CLI Command Reference.

## send-command

Il seguente esempio di codice mostra come utilizzare `send-command`.

### AWS CLI

Esempio 1: eseguire un comando su una o più istanze remote

L'`send-command` seguente esegue un `echo` comando su un'istanza di destinazione.

```
aws ssm send-command \  
  --document-name "AWS-RunShellScript" \  
  --parameters 'commands=["echo HelloWorld"]' \  
  --targets "Key=instanceids,Values=i-1234567890abcdef0" \  
  --comment "echo HelloWorld"
```

Output:

```
{  
  "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
    }
  }
}

```

Per ulteriori informazioni, vedere [Running Commands Using Systems Manager Run Command](#) nella AWS Systems Manager User Guide.

Esempio 2: per ottenere informazioni IP su un'istanza

L'end-comandesempio seguente recupera le informazioni IP relative a un'istanza.

```

aws ssm send-command \
  --instance-ids "i-1234567890abcdef0" \
  --document-name "AWS-RunShellScript" \
  --comment "IP config" \
  --parameters "commands=ifconfig"

```

Vedi l'esempio 1 per un esempio di output.

Per ulteriori informazioni, vedere [Running Commands Using Systems Manager Run Command](#) nella AWS Systems Manager User Guide.

Esempio 3: eseguire un comando su istanze con tag specifici

L'end-commandesempio seguente esegue un comando sulle istanze che hanno la chiave tag "ENV" e il valore «Dev».

```
aws ssm send-command \  
  --targets "Key=tag:ENV,Values=Dev" \  
  --document-name "AWS-RunShellScript" \  
  --parameters "commands=ifconfig"
```

Vedi l'esempio 1 per un output di esempio.

Per ulteriori informazioni, vedere [Running Commands Using Systems Manager Run Command](#) nella AWS Systems Manager User Guide.

Esempio 4: eseguire un comando che invia SNS notifiche

L'end-commandesempio seguente esegue un comando che invia SNS notifiche per tutti gli eventi di notifica e il tipo di Command notifica.

```
aws ssm send-command \  
  --instance-ids "i-1234567890abcdef0" \  
  --document-name "AWS-RunShellScript" \  
  --comment "IP config" \  
  --parameters "commands=ifconfig" \  
  --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"
```

Per un esempio di output, vedere l'esempio 1.

Per ulteriori informazioni, vedere [Running Commands Using Systems Manager Run Command](#) nella AWS Systems Manager User Guide.

Esempio 5: eseguire un comando con output su S3 e CloudWatch

L'end-commandesempio seguente esegue un comando che invia i dettagli del comando a un bucket S3 e a un gruppo di log Logs. CloudWatch

```
aws ssm send-command \  
  --instance-ids "i-1234567890abcdef0" \  
  --document-name "AWS-RunShellScript" \  
  --comment "IP config" \  
  --parameters "commands=ifconfig"
```



```
--parameters "commands=ifconfig" \  
--output-s3-bucket-name "s3-bucket-name" \  
--output-s3-key-prefix "runcommand" \  
--cloud-watch-output-  
config "CloudWatchOutputEnabled=true,CloudWatchLogGroupName=CWLGroupName"
```

Vedi l'esempio 1 per un output di esempio.

Per ulteriori informazioni, vedere [Running Commands Using Systems Manager Run Command](#) nella AWS Systems Manager User Guide.

Esempio 6: eseguire comandi su più istanze con tag diversi

L'send-commandesempio seguente esegue un comando su istanze con due chiavi e valori di tag diversi.

```
aws ssm send-command \  
--document-name "AWS-RunPowerShellScript" \  
--parameters commands=["echo helloWorld"] \  
--targets Key=tag:Env,Values=Dev Key=tag:Role,Values=WebServers
```

Vedi l'esempio 1 per un esempio di output.

Per ulteriori informazioni, vedere [Running Commands Using Systems Manager Run Command](#) nella AWS Systems Manager User Guide.

Esempio 7: Per indirizzare più istanze con la stessa chiave di tag

L'send-commandesempio seguente esegue un comando su istanze che hanno la stessa chiave di tag ma con valori diversi.

```
aws ssm send-command \  
--document-name "AWS-RunPowerShellScript" \  
--parameters commands=["echo helloWorld"] \  
--targets Key=tag:Env,Values=Dev,Test
```

Vedi l'esempio 1 per un output di esempio.

Per ulteriori informazioni, vedere [Running Commands Using Systems Manager Run Command](#) nella AWS Systems Manager User Guide.

## Esempio 8: eseguire un comando che utilizza un documento condiviso

L'`send-command` seguente esegue un documento condiviso su un'istanza di destinazione.

```
aws ssm send-command \  
  --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument" \  
  --targets "Key=instanceids,Values=i-1234567890abcdef0"
```

Vedi l'esempio 1 per un output di esempio.

Per ulteriori informazioni, vedere [Utilizzo di SSM documenti condivisi](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [SendCommand](#) in AWS CLI Command Reference.

## `start-associations-once`

Il seguente esempio di codice mostra come utilizzare `start-associations-once`.

### AWS CLI

Per eseguire un'associazione immediatamente e solo una volta

L'`start-associations-once` seguente esegue l'associazione specificata immediatamente e solo una volta. Se il comando va a buon fine, non viene generato output.

```
aws ssm start-associations-once \  
  --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Visualizzazione della cronologia delle associazioni](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [StartAssociationsOnce](#) in AWS CLI Command Reference.

## `start-automation-execution`

Il seguente esempio di codice mostra come utilizzare `start-automation-execution`.

## AWS CLI

Esempio 1: Eseguire un documento di automazione

L'`start-automation-execution` seguente esegue un documento di automazione.

```
aws ssm start-automation-execution \  
  --document-name "AWS-UpdateLinuxAmi" \  
  --parameters "AutomationAssumeRole=arn:aws:iam::123456789012:role/  
SSMAutomationRole,SourceAmiId=ami-EXAMPLE,IamInstanceProfileName=EC2InstanceRole"
```

Output:

```
{  
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"  
}
```

Per ulteriori informazioni, vedere [Esecuzione manuale di un flusso di lavoro di automazione](#) nella Guida per l'utente di AWS Systems Manager.

Esempio 2: Per eseguire un documento di automazione condiviso

L'`start-automation-execution` seguente esegue un documento di automazione condiviso.

```
aws ssm start-automation-execution \  
  --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument"
```

Output:

```
{  
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"  
}
```

Per ulteriori informazioni, vedere [Utilizzo di SSM documenti condivisi](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [StartAutomationExecution](#) in AWS CLI Command Reference.

## **start-change-request-execution**

Il seguente esempio di codice mostra come utilizzare `start-change-request-execution`.

## AWS CLI

Esempio 1: per avviare una richiesta di modifica

L'`start-change-request-execution` seguente avvia una richiesta di modifica con opzioni minime specificate.

```
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"
```

Output:

```
{  
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"  
}
```

Esempio 2: per avviare una richiesta di modifica utilizzando un JSON file esterno

L'`start-automation-execution` seguente avvia una richiesta di modifica con più opzioni specificate in un JSON file.

```
aws ssm start-change-request-execution \  
  --cli-input-json file://MyChangeRequest.json
```

Contenuto di `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\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\n\nThis document has no outputs \n"
}

```

### Output:

```

{
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"
}

```

Per ulteriori informazioni, vedere [Creazione di richieste di modifica](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [StartChangeRequestExecution](#) in AWS CLI Command Reference.

## start-session

Il seguente esempio di codice mostra come utilizzare `start-session`.

### AWS CLI

Esempio 1: per avviare una sessione di Session Manager

Questo `start-session` esempio stabilisce una connessione con un'istanza per una sessione di Session Manager. Si noti che questo comando interattivo richiede l'installazione del plug-in Session Manager sul computer client che effettua la chiamata.

```
aws ssm start-session \  
  --target "i-1234567890abcdef0"
```

Output:

```
Starting session with SessionId: Jane-Roe-07a16060613c408b5
```

Esempio 2: Per avviare una sessione di Session Manager utilizzando SSH

Questo `start-session` esempio stabilisce una connessione con un'istanza per una sessione di Session Manager utilizzando SSH. Si noti che questo comando interattivo richiede l'installazione del plug-in Session Manager sul computer client che effettua la chiamata e che il comando utilizza l'utente predefinito sull'istanza, ad esempio per EC2 le istanze `ec2-user` per Linux.

```
ssh -i /path/my-key-pair.pem ec2-user@i-02573cafcfEXAMPLE
```

Output:

```
Starting session with SessionId: ec2-user-07a16060613c408b5
```

Per ulteriori informazioni, vedere [Avvio di una sessione](#) e [installazione del plug-in Session Manager AWS CLI nella Guida per l'utente](#) di AWS Systems Manager.

- Per API i dettagli, vedere [StartSession](#) in AWS CLI Command Reference.

## stop-automation-execution

Il seguente esempio di codice mostra come utilizzare `stop-automation-execution`.

### AWS CLI

Per interrompere l'esecuzione di un'automazione

L'`stop-automation-execution` esempio seguente interrompe un documento di automazione.

```
aws ssm stop-automation-execution
  --automation-execution-id "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Esecuzione manuale di un flusso di lavoro di automazione](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [StopAutomationExecution](#) in AWS CLI Command Reference.

## terminate-session

Il seguente esempio di codice mostra come utilizzare `terminate-session`.

### AWS CLI

Per terminare una sessione di Session Manager

Questo `terminate-session` esempio termina definitivamente una sessione creata dall'utente «Shirley-Rodriguez» e chiude la connessione dati tra il client Session Manager e l'agente sull'istanza. SSM

```
aws ssm terminate-session \
  --session-id "Shirley-Rodriguez-07a16060613c408b5"
```

Output:

```
{
  "SessionId": "Shirley-Rodriguez-07a16060613c408b5"
```

```
}
```

Per ulteriori informazioni, vedere [Terminare una sessione](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [TerminateSession](#) in AWS CLI Command Reference.

## **unlabel-parameter-version**

Il seguente esempio di codice mostra come utilizzare `unlabel-parameter-version`.

### AWS CLI

Per eliminare le etichette dei parametri

L'`unlabel-parameter-version` seguente elimina le etichette specificate dalla versione del parametro specificata.

```
aws ssm unlabel-parameter-version \
  --name "parameterName" \
  --parameter-version "version" \
  --labels "label_1" "label_2" "label_3"
```

Output:

```
{
  "RemovedLabels": [
    "label_1"
    "label_2"
    "label_3"
  ],
  "InvalidLabels": []
}
```

Per ulteriori informazioni, vedere [Delete parameter labels \(AWS CLI\)](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [UnlabelParameterVersion](#) in AWS CLI Command Reference.

## **update-association-status**

Il seguente esempio di codice mostra come utilizzare `update-association-status`.



## AWS CLI

Per aggiornare lo stato dell'associazione

L'update-association-statusesempio seguente aggiorna lo stato dell'associazione tra un'istanza e un documento.

```
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"
```

Output:

```
{
  "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
}
}

```

Per ulteriori informazioni, vedere [Utilizzo delle associazioni in Systems Manager](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [UpdateAssociationStatus](#) in AWS CLI Command Reference.

## update-association

Il seguente esempio di codice mostra come utilizzare `update-association`.

### AWS CLI

Esempio 1: Per aggiornare un'associazione di documenti

L'`update-association` seguente aggiorna un'associazione con una nuova versione del documento.

```

aws ssm update-association \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
  --document-version "$LATEST"

```

Output:

```

{
  "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
}
}

```

Per ulteriori informazioni, vedere [Modifica e creazione di una nuova versione di un'associazione](#) nella Guida per l'utente di AWS Systems Manager.

Esempio 2: aggiornare l'espressione di pianificazione di un'associazione

L'update-associationesempio seguente aggiorna l'espressione di pianificazione per l'associazione specificata.

```

aws ssm update-association \
  --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
  --schedule-expression "cron(0 0 0/4 1/1 * ? *)"

```

Output:

```

{
  "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
}
}

```

Per ulteriori informazioni, vedere [Modifica e creazione di una nuova versione di un'associazione](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [UpdateAssociation](#) in AWS CLI Command Reference.

### update-document-default-version

Il seguente esempio di codice mostra come utilizzare `update-document-default-version`.

#### AWS CLI

Per aggiornare la versione predefinita di un documento

L'esempio seguente aggiorna la versione predefinita di un documento Systems Manager.

```

aws ssm update-document-default-version \
  --name "Example" \
  --document-version "2"

```

Output:

```

{
  "Description": {
    "Name": "Example",
    "DefaultVersion": "2"
  }
}

```

Per ulteriori informazioni, vedere [Writing SSM Document Content](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [UpdateDocumentDefaultVersion](#) in AWS CLI Command Reference.

## update-document-metadata

Il seguente esempio di codice mostra come utilizzare `update-document-metadata`.

### AWS CLI

Esempio: approvare la versione più recente di un modello di modifica

Quanto segue `update-document-metadata` fornisce un'approvazione per la versione più recente di un modello di modifica che è stato inviato per la revisione.

```
aws ssm update-document-metadata \  
  --name MyChangeManagerTemplate \  
  --document-reviews 'Action=Approve, Comment=[{Type=Comment, Content=Approved!}]'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Revisione e approvazione o rifiuto dei modelli di modifica](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [UpdateDocumentMetadata](#) in AWS CLI Command Reference.

## update-document

Il seguente esempio di codice mostra come utilizzare `update-document`.

### AWS CLI

Per creare una nuova versione di un documento

L'`update-document` esempio seguente crea una nuova versione di un documento quando viene eseguito su un computer Windows. Il documento specificato da `--document` deve essere in JSON formato. Nota che `file://` deve essere referenziato seguito dal percorso del file di contenuto. Poiché all'inizio del `--document-version` parametro, `$` in Windows è necessario racchiudere il valore tra virgolette doppie. Su Linux, macOS o su PowerShell richiesta, è necessario racchiudere il valore tra virgolette singole.

Versione per Windows:

```
aws ssm update-document \  
  --document file://C:\path\to\document.json \  
  --document-version $
```

```
--name "RunShellScript" \  
--content "file://RunShellScript.json" \  
--document-version "$LATEST"
```

Versione Linux/Mac:

```
aws ssm update-document \  
  --name "RunShellScript" \  
  --content "file://RunShellScript.json" \  
  --document-version '$LATEST'
```

Output:

```
{  
  "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",  
    "CreateDate": 1487899655.152,  
    "Owner": "809632081692",  
    "SchemaVersion": "2.0",  
    "DefaultVersion": "1",  
    "LatestVersion": "2",  
    "Description": "Run an updated script"  
  }  
}
```

- Per i API dettagli, vedere [UpdateDocument](#) in Command Reference.AWS CLI

## update-maintenance-window-target

Il seguente esempio di codice mostra come utilizzare `update-maintenance-window-target`.

### AWS CLI

Per aggiornare la destinazione di una finestra di manutenzione

L'`update-maintenance-window-target` seguente aggiorna solo il nome di una destinazione della finestra di manutenzione.

```
aws ssm update-maintenance-window-target \
  --window-id "mw-0c5ed765acEXAMPLE" \
  --window-target-id "57e8344e-fe64-4023-8191-6bf05EXAMPLE" \
  --name "NewName" \
  --no-replace
```

Output:

```
{
  "Description": "",
  "OwnerInformation": "",
  "WindowTargetId": "57e8344e-fe64-4023-8191-6bf05EXAMPLE",
  "WindowId": "mw-0c5ed765acEXAMPLE",
  "Targets": [
    {
      "Values": [
        "i-1234567890EXAMPLE"
      ],
      "Key": "InstanceIds"
    }
  ],
  "Name": "NewName"
}
```

Per ulteriori informazioni, vedere [Update a Maintenance Window \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [UpdateMaintenanceWindowTarget](#) in AWS CLI Command Reference.

## update-maintenance-window-task

Il seguente esempio di codice mostra come utilizzare `update-maintenance-window-task`.

## AWS CLI

Per aggiornare un'operazione della finestra di manutenzione

L'update-maintenance-window-taskesempio seguente aggiorna il ruolo di servizio per un'attività della finestra di manutenzione.

```
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"
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, vedere [Update a Maintenance Window \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [UpdateMaintenanceWindowTask](#) in AWS CLI Command Reference.

## update-maintenance-window

Il seguente esempio di codice mostra come utilizzare `update-maintenance-window`.

### AWS CLI

Esempio 1: Per aggiornare una finestra di manutenzione

L'`update-maintenance-window` esempio seguente aggiorna il nome di una finestra di manutenzione.

```
aws ssm update-maintenance-window \  
  --window-id "mw-1a2b3c4d5e6f7g8h9" \  
  --name "My-Renamed-MW"
```

Output:

```
{  
  "Cutoff": 1,  
  "Name": "My-Renamed-MW",  
  "Schedule": "cron(0 16 ? * TUE *)",  
  "Enabled": true,  
  "AllowUnassociatedTargets": true,  
  "WindowId": "mw-1a2b3c4d5e6f7g8h9",  
  "Duration": 4  
}
```

Esempio 2: disabilitare una finestra di manutenzione

L'`update-maintenance-window` esempio seguente disattiva una finestra di manutenzione.

```
aws ssm update-maintenance-window \  
  --window-id "mw-1a2b3c4d5e6f7g8h9" \  
  --name "My-Renamed-MW" \  
  --enable
```

```
--window-id "mw-1a2b3c4d5e6f7g8h9" \  
--no-enabled
```

Esempio 3: Per abilitare una finestra di manutenzione

L'update-maintenance-windowesempio seguente abilita una finestra di manutenzione.

```
aws ssm update-maintenance-window \  
--window-id "mw-1a2b3c4d5e6f7g8h9" \  
--enabled
```

Per ulteriori informazioni, vedere [Update a Maintenance Window \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [UpdateMaintenanceWindow](#) in AWS CLI Command Reference.

## update-managed-instance-role

Il seguente esempio di codice mostra come utilizzareupdate-managed-instance-role.

AWS CLI

Per aggiornare il IAM ruolo di un'istanza gestita

L'update-managed-instance-rolesempio seguente aggiorna il profilo di IAM istanza di un'istanza gestita.

```
aws ssm update-managed-instance-role \  
--instance-id "mi-08ab247cdfEXAMPLE" \  
--iam-role "ExampleRole"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Fase 4: Creazione di un profilo di IAM istanza per Systems Manager](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [UpdateManagedInstanceRole](#) in AWS CLI Command Reference.

## update-ops-item

Il seguente esempio di codice mostra come utilizzareupdate-ops-item.

## AWS CLI

Per aggiornare un OpsItem

L'update-ops-item esempio seguente aggiorna la descrizione, la priorità e la categoria di un OpsItem. Inoltre, il comando specifica un SNS argomento in cui vengono inviate le notifiche quando questo OpsItem viene modificato o modificato.

```
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"
```

Output:

```
This command produces no output.
```

Per ulteriori informazioni, vedere [Working with OpsItems](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [UpdateOpsItem](#) in AWS CLI Command Reference.

## update-patch-baseline

Il seguente esempio di codice mostra come utilizzare update-patch-baseline.

### AWS CLI

Esempio 1: Per aggiornare una patch di base

L'update-patch-baseline esempio seguente aggiunge le due patch specificate come rifiutate e una patch approvata alla baseline della patch specificata.

```
aws ssm update-patch-baseline \  
  --baseline-id "pb-0123456789abcdef0" \  
  --rejected-patches "KB2032276" "MS10-048" \  
  --approved-patches "KB2124261"
```

Output:

```
{
  "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": []
}
```

## Esempio 2: rinominare una patch di base

L'update-patch-baselineesempio seguente rinomina la linea di base della patch specificata.

```
aws ssm update-patch-baseline \  
  --baseline-id "pb-0713accee01234567" \  
  --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"
```

Per ulteriori informazioni, vedere Update or Delete a Patch Baseline < <https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-baseline-update-or-delete.html>> nella Systems AWS Manager User Guide.

- Per i API dettagli, vedere in Command Reference. [UpdatePatchBaseline](#) AWS CLI

## update-resource-data-sync

Il seguente esempio di codice mostra come utilizzareupdate-resource-data-sync.

AWS CLI

Per aggiornare la sincronizzazione dei dati di una risorsa

L'update-resource-data-syncesempio seguente aggiorna la sincronizzazione dei dati di una SyncFromSource risorsa.

```
aws ssm update-resource-data-sync \  
  --sync-name exampleSync \  
  --sync-type SyncFromSource \  
  --sync-source '{"SourceType": "SingleAccountMultiRegions", "SourceRegions": ["us-east-1", "us-west-2"]}'
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Configurazione di Systems Manager Explorer per la visualizzazione dei dati da più account e regioni](#) nella Guida per l'utente di AWS Systems Manager.

- Per API i dettagli, vedere [UpdateResourceDataSync](#) in AWS CLI Command Reference.

## update-service-setting

Il seguente esempio di codice mostra come utilizzareupdate-service-setting.

## AWS CLI

Per aggiornare l'impostazione del servizio per il throughput di Parameter Store

L'update-service-setting esempio seguente aggiorna l'impostazione corrente del servizio per la velocità effettiva di Parameter Store nella regione specificata per utilizzare una maggiore velocità effettiva.

```
aws ssm update-service-setting \  
  --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-  
store/high-throughput-enabled \  
  --setting-value true
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Increasing Parameter Store Throughput](#) nella AWS Systems Manager User Guide.

- Per API i dettagli, vedere [UpdateServiceSetting](#) in AWS CLI Command Reference.

## Esempi di utilizzo di Amazon Textract AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Textract. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **analyze-document**

Il seguente esempio di codice mostra come utilizzare `analyze-document`.

## AWS CLI

Per analizzare il testo in un documento

L'analyze-documentesempio seguente mostra come analizzare il testo in un documento.

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
```

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": [
      "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
}
}
```

Per ulteriori informazioni, consulta [Analyzing Document Text with Amazon Textract](#) nella [Amazon Textract Developers Guide](#)

- Per API i dettagli, consulta [Command `AnalyzeDocument` Reference AWS CLI](#) .

## **detect-document-text**

Il seguente esempio di codice mostra come utilizzare `detect-document-text`.

### AWS CLI

Per rilevare il testo in un documento

L'esempio seguente mostra come rilevare il testo in un documento. `detect-document-text`

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
```

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": [  

```

```
        "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": "He llo,",
  "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
    }
  }
}
```

Per ulteriori informazioni, consulta [Detecting Document Text with Amazon Textract](#) nella [Amazon Textract Developers Guide](#)

- Per API i dettagli, consulta [Command DetectDocumentTextReference AWS CLI](#) .

## get-document-analysis

Il seguente esempio di codice mostra come utilizzare `get-document-analysis`.

### AWS CLI

Per ottenere i risultati dell'analisi asincrona del testo di un documento composto da più pagine

L'`get-document-analysis` esempio seguente mostra come ottenere i risultati dell'analisi asincrona del testo di un documento composto da più pagine.

```
aws textract get-document-analysis \
  --job-id df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b \
  --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": [
            "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"
}

```

Per ulteriori informazioni, consulta la sezione Rilevamento e analisi del testo nei documenti multipagina nella Amazon Textract Developers Guide

- Per i API dettagli, consulta [GetDocumentAnalysis](#) Command Reference.AWS CLI

## get-document-text-detection

Il seguente esempio di codice mostra come utilizzare `get-document-text-detection`.

### AWS CLI

Per ottenere i risultati del rilevamento asincrono del testo in un documento composto da più pagine

L'esempio seguente mostra come ottenere i risultati del rilevamento asincrono del testo in un documento composto da più pagine.

```
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  
          }  
        ]  
      }  
    }  
  ]  
}
```

```

        "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/
muqc96S4o5NzDqehhXvhkodMyV050JGyms5lsrCxibWJw==",
"DocumentMetadata": {
  "Pages": 1
},
"JobStatus": "SUCCEEDED"
}

```

Per ulteriori informazioni, consulta la sezione Rilevamento e analisi del testo nei documenti multipagina nella Amazon Textract Developers Guide

- Per i API dettagli, consulta [GetDocumentTextDetection](#) Command Reference.AWS CLI

## start-document-analysis

Il seguente esempio di codice mostra come utilizzare `start-document-analysis`.

### AWS CLI

Per iniziare ad analizzare il testo in un documento composto da più pagine

L'`start-document-analysis` esempio seguente mostra come avviare l'analisi asincrona del testo in un documento composto da più pagine.

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\"]]\" \  
  --feature-types [\"TABLES\", \"FORMS\"] \  
  --region region-name \  
  --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleArn"
```

Output:

```
{  
  "JobId": "df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b"  
}
```

Per ulteriori informazioni, consulta la sezione Rilevamento e analisi del testo nei documenti multipagina nella Amazon Textract Developers Guide

- Per i API dettagli, consulta [StartDocumentAnalysis](#) Command Reference.AWS CLI

## start-document-text-detection

Il seguente esempio di codice mostra come utilizzare `start-document-text-detection`.

AWS CLI

Per iniziare a rilevare il testo in un documento composto da più pagine

L'`start-document-text-detection` esempio seguente mostra come avviare il rilevamento asincrono del testo in un documento composto da più pagine.

Linux/macOS:

```
aws textract start-document-text-detection \  

```



```
--document-location '{"S3Object":{"Bucket":"bucket","Name":"document"}}' \  
--notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleARN"
```

Windows:

```
aws textract start-document-text-detection \  
  --document-location "{\"S3Object\":{\"Bucket\":\"bucket\",\"Name\":\"document\  
  \"}}" \  
  --region region-name \  
  --notification-channel "SNSTopicArn=arn:snsTopic,RoleArn=roleArn"
```

Output:

```
{  
  "JobId": "57849a3dc627d4df74123dca269d69f7b89329c870c65bb16c9fd63409d200b9"  
}
```

Per ulteriori informazioni, consulta la sezione Rilevamento e analisi del testo nei documenti multipagina nella Amazon Textract Developers Guide

- Per i API dettagli, consulta [StartDocumentTextDetection](#) Command Reference.AWS CLI

## Esempi di Amazon Transcribe utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Transcribe. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come chiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

## Azioni

### **create-language-model**

Il seguente esempio di codice mostra come utilizzare `create-language-model`.

#### AWS CLI

Esempio 1: creare un modello linguistico personalizzato utilizzando sia i dati di addestramento che quelli di ottimizzazione.

L'esempio seguente crea un modello linguistico personalizzato. È possibile utilizzare un modello linguistico personalizzato per migliorare le prestazioni di trascrizione per settori quali quello legale, alberghiero, finanziario e assicurativo. Per il codice della lingua, inserisci un codice lingua valido. Per `base-model-name`, specificate un modello base più adatto alla frequenza di campionamento dell'audio che desiderate trascrivere con il vostro modello linguistico personalizzato. Per `model-name`, specificate il nome con cui desiderate chiamare il modello linguistico personalizzato.

```
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"
```

Output:

```
{
  "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"
```

```
}
```

Per ulteriori informazioni, consulta [Modelli linguistici personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

Esempio 2: creare un modello linguistico personalizzato utilizzando solo dati di addestramento.

L'esempio `create-language-model` seguente trascrive il file audio. Puoi utilizzare un modello linguistico personalizzato per migliorare le prestazioni di trascrizione per settori come quello legale, alberghiero, finanziario e assicurativo. Per il codice della lingua, inserisci un codice lingua valido. Per `base-model-name`, specificate un modello base più adatto alla frequenza di campionamento dell'audio che desiderate trascrivere con il vostro modello linguistico personalizzato. Per `model-name`, specificate il nome con cui desiderate chiamare il modello linguistico personalizzato.

```
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"
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Modelli linguistici personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per API i dettagli, vedere [CreateLanguageModel](#) in AWS CLI Command Reference.

## create-medical-vocabulary

Il seguente esempio di codice mostra come utilizzare `create-medical-vocabulary`.

### AWS CLI

Per creare un vocabolario medico personalizzato

L'esempio `create-medical-vocabulary` seguente crea un vocabolario personalizzato. Per creare un vocabolario personalizzato, devi aver creato un file di testo con tutti i termini che desideri trascrivere in modo più accurato. Per `vocabulary-file-uri`, specifica Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon URI S3) di quel file di testo. Per `language-code`, specifica un codice di lingua corrispondente alla lingua del vocabolario personalizzato. Per `vocabulary-name`, specifica come vuoi denominare il vocabolario personalizzato.

```
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
```

Output:

```
{  
  "VocabularyName": "cli-medical-vocab-example",  
  "LanguageCode": "language-code",  
  "VocabularyState": "PENDING"  
}
```

Per ulteriori informazioni, consulta [Vocabolari medici personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per API i dettagli, consulta [CreateMedicalVocabulary AWS CLI](#) Command Reference.

## create-vocabulary-filter

Il seguente esempio di codice mostra come utilizzare `create-vocabulary-filter`.

### AWS CLI

Per creare un filtro per il vocabolario

L'`create-vocabulary-filter` seguente crea un filtro per il vocabolario che utilizza un file di testo contenente un elenco di parole che non si desidera vengano visualizzate in una trascrizione. Per `language-code`, specificate il codice della lingua corrispondente alla lingua del filtro del vocabolario. Per `vocabulary-filter-file-uri`, specifica Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon URI S3) del file di testo. Per `vocabulary-filter-name`, specifica il nome del filtro del vocabolario.

```
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
```

Output:

```
{  
  "VocabularyFilterName": "cli-vocabulary-filter-example",  
  "LanguageCode": "language-code"  
}
```

Per ulteriori informazioni, consulta [Filtrare le parole indesiderate](#) nella Amazon Transcribe Developer Guide.

- Per API i dettagli, consulta Command [CreateVocabularyFilter](#) Reference AWS CLI .

## **create-vocabulary**

Il seguente esempio di codice mostra come utilizzare `create-vocabulary`.

### AWS CLI

#### Creazione di un vocabolario personalizzato

L'esempio `create-vocabulary` seguente crea un vocabolario personalizzato. Per creare un vocabolario personalizzato, devi aver creato un file di testo con tutti i termini che desideri trascrivere in modo più accurato. Per `vocabulary-file-uri`, specifica Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon URI S3) di quel file di testo. Per `language-code`, specifica un codice di lingua corrispondente alla lingua del vocabolario personalizzato. Per `vocabulary-name`, specifica come vuoi denominare il vocabolario personalizzato.

```
aws transcribe create-vocabulary \  
  --language-code language-code \  
  --vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/vocabulary-filter.txt \  
  --vocabulary-name cli-vocabulary-filter-example
```

```
--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
```

Output:

```
{  
  "VocabularyName": "cli-vocab-example",  
  "LanguageCode": "language-code",  
  "VocabularyState": "PENDING"  
}
```

Per ulteriori informazioni, consulta [Vocabolari personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per API i dettagli, consulta [CreateVocabulary AWS CLI](#) Command Reference.

## delete-language-model

Il seguente esempio di codice mostra come utilizzare `delete-language-model`.

AWS CLI

Per eliminare un modello linguistico personalizzato

L'esempio seguente elimina un modello linguistico personalizzato.

```
aws transcribe delete-language-model \  
  --model-name model-name
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Modelli linguistici personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per API i dettagli, vedere [DeleteLanguageModel](#) in AWS CLI Command Reference.

## delete-medical-transcription-job

Il seguente esempio di codice mostra come utilizzare `delete-medical-transcription-job`.

## AWS CLI

Eliminazione di un processo di trascrizione medica

L'esempio `delete-medical-transcription-job` seguente elimina un processo di trascrizione medica.

```
aws transcribe delete-medical-transcription-job \  
  --medical-transcription-job-name medical-transcription-job-name
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [DeleteMedicalTranscriptionJob](#) la Amazon Transcribe Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteMedicalTranscriptionJob](#) Reference.

## `delete-medical-vocabulary`

Il seguente esempio di codice mostra come utilizzare `delete-medical-vocabulary`.

## AWS CLI

Per eliminare un vocabolario medico personalizzato

L'esempio `delete-medical-vocabulary` seguente elimina un vocabolario medico personalizzato. Per `vocabulary-name`, specifica il nome del vocabolario medico personalizzato.

```
aws transcribe delete-vocabulary \  
  --vocabulary-name medical-custom-vocabulary-name
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Vocabolari medici personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per i dettagli, vedere in Command Reference. API [DeleteMedicalVocabulary](#) AWS CLI

## `delete-transcription-job`

Il seguente esempio di codice mostra come utilizzare `delete-transcription-job`.

## AWS CLI

Eliminazione di un processo di trascrizione

L'esempio `delete-transcription-job` seguente elimina uno dei processi di trascrizione.

```
aws transcribe delete-transcription-job \  
  --transcription-job-name your-transcription-job
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [DeleteTranscriptionJob](#) la Amazon Transcribe Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteTranscriptionJob](#) Reference.

## `delete-vocabulary-filter`

Il seguente esempio di codice mostra come utilizzare `delete-vocabulary-filter`.

## AWS CLI

Per eliminare un filtro del vocabolario

L'`delete-vocabulary-filter` esempio seguente elimina un filtro del vocabolario.

```
aws transcribe delete-vocabulary-filter \  
  --vocabulary-filter-name vocabulary-filter-name
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Filtrare le parole indesiderate](#) nella Amazon Transcribe Developer Guide.

- Per API i dettagli, consulta Command [DeleteVocabularyFilter](#) Reference AWS CLI .

## `delete-vocabulary`

Il seguente esempio di codice mostra come utilizzare `delete-vocabulary`.

## AWS CLI

Eliminazione di un vocabolario personalizzato



L'esempio `delete-vocabulary` seguente elimina un vocabolario personalizzato.

```
aws transcribe delete-vocabulary \  
  --vocabulary-name vocabulary-name
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Vocabolari personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per API i dettagli, vedere [DeleteVocabulary](#) in AWS CLI Command Reference.

## describe-language-model

Il seguente esempio di codice mostra come utilizzare `describe-language-model`.

### AWS CLI

Per ottenere informazioni su uno specifico modello linguistico personalizzato

L'esempio seguente ottiene informazioni su uno specifico modello linguistico personalizzato. Ad esempio, sotto `BaseModelName` puoi vedere se il tuo modello è stato addestrato utilizzando un `WideBand` modello `NarrowBand` o. I modelli linguistici personalizzati con un modello `NarrowBand` base possono trascrivere l'audio con una frequenza di campionamento inferiore a 16. kHz I modelli linguistici che utilizzano un modello `WideBand` base possono trascrivere l'audio con una frequenza di campionamento superiore a 16. kHz Il parametro `S3Uri` indica il prefisso Amazon S3 che hai utilizzato per accedere ai dati di formazione per creare il modello linguistico personalizzato.

```
aws transcribe describe-language-model \  
  --model-name cli-clm-example
```

Output:

```
{  
  "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"
    }
  }
}
```

Per ulteriori informazioni, consulta [Modelli linguistici personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per i API dettagli, consulta [DescribeLanguageModel](#) Command Reference.AWS CLI

## get-medical-transcription-job

Il seguente esempio di codice mostra come utilizzare `get-medical-transcription-job`.

### AWS CLI

Per ottenere informazioni su uno specifico lavoro di trascrizione medica

L'`get-medical-transcription-job`esempio seguente ottiene informazioni su uno specifico lavoro di trascrizione medica. Per accedere ai risultati della trascrizione, utilizzate il parametro `TranscriptFileUri` Se hai abilitato funzionalità aggiuntive per il processo di trascrizione, puoi visualizzarle nell'oggetto Impostazioni. Il parametro `Specialità` mostra la specialità medica del fornitore. Il parametro `Type` indica se il discorso nel processo di trascrizione è tratto da una conversazione medica o da un dettato medico.

```
aws transcribe get-medical-transcription-job \
  --medical-transcription-job-name vocabulary-dictation-medical-transcription-job
```

Output:

```
{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Batch Transcription nella Amazon Transcribe Developer Guide](#).

- Per i API dettagli, consulta [GetMedicalTranscriptionJob](#) Command Reference.AWS CLI

## get-medical-vocabulary

Il seguente esempio di codice mostra come utilizzare `get-medical-vocabulary`.

### AWS CLI

Per ottenere informazioni su un vocabolario medico personalizzato

L'`get-medical-vocabulary` esempio seguente ottiene informazioni su un vocabolario medico personalizzato. È possibile utilizzare il `VocabularyState` parametro per visualizzare lo stato di elaborazione del vocabolario. Se lo è `READY`, puoi usarlo nell'operazione.

`StartMedicalTranscriptionJob` :

```

aws transcribe get-medical-vocabulary \
  --vocabulary-name medical-vocab-example

```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Vocabolari medici personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per API i dettagli, vedere [GetMedicalVocabulary](#) in AWS CLI Command Reference.

## get-transcription-job

Il seguente esempio di codice mostra come utilizzare `get-transcription-job`.

### AWS CLI

Ottenere informazioni su un processo di trascrizione specifico

L'esempio `get-transcription-job` seguente ottiene informazioni su un processo di trascrizione specifico. Per accedere ai risultati della trascrizione, utilizzate il `TranscriptFileUri` parametro. Utilizzate il `MediaFileUri` parametro per vedere quale file audio avete trascritto con questo lavoro. Puoi usare l'oggetto `Settings` per visualizzare le funzionalità opzionali che hai abilitato nel processo di trascrizione.

```
aws transcribe get-transcription-job \
  --transcription-job-name your-transcription-job
```

Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Getting Started \(AWS Command Line Interface\)](#) nella Amazon Transcribe Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetTranscriptionJobReference](#).

## get-vocabulary-filter

Il seguente esempio di codice mostra come utilizzare `get-vocabulary-filter`.

### AWS CLI

Per ottenere informazioni su un filtro del vocabolario

L'`get-vocabulary-filter` esempio seguente ottiene informazioni su un filtro del vocabolario. È possibile utilizzare il `DownloadUri` parametro per ottenere l'elenco delle parole utilizzate per creare il filtro del vocabolario.

```
aws transcribe get-vocabulary-filter \
  --vocabulary-filter-name testFilter
```

Output:

```
{
  "VocabularyFilterName": "testFilter",
  "LanguageCode": "language-code",
```

```
"LastModifiedTime": "2020-05-07T22:39:32.147000+00:00",  
"DownloadUri": "https://Amazon-S3-location-to-download-your-vocabulary-filter"  
}
```

Per ulteriori informazioni, consulta [Filtrare le parole indesiderate](#) nella Amazon Transcribe Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [GetVocabularyFilter](#)Reference.

## get-vocabulary

Il seguente esempio di codice mostra come utilizzare `get-vocabulary`.

### AWS CLI

Ottenere informazioni su un vocabolario personalizzato

L'esempio `get-vocabulary` seguente ottiene informazioni su un vocabolario personalizzato creato in precedenza.

```
aws transcribe get-vocabulary \  
  --vocabulary-name cli-vocab-1
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Vocabolari personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per API i dettagli, vedere [GetVocabulary](#) in AWS CLI Command Reference.

## list-language-models

Il seguente esempio di codice mostra come utilizzare `list-language-models`.

## AWS CLI

Per elencare i modelli linguistici personalizzati

L'`list-language-models` seguente elenca i modelli linguistici personalizzati associati all'AWS account e alla regione. Puoi utilizzare i `TuningDataS3Uri` parametri `S3Uri` and per trovare i prefissi Amazon S3 che hai usato come dati di allenamento o dati di ottimizzazione. Ti `BaseModelName` dice se hai usato un `WideBand` modello o un `NarrowBand` modello per creare un modello linguistico personalizzato. È possibile trascrivere l'audio con una frequenza di campionamento inferiore a 16 kHz con un modello linguistico personalizzato utilizzando un modello `NarrowBand` base. È possibile trascrivere l'audio 16 kHz o superiore con un modello linguistico personalizzato utilizzando un modello base. `WideBand` Il `ModelStatus` parametro mostra se è possibile utilizzare il modello linguistico personalizzato in un lavoro di trascrizione. Se il valore è `COMPLETED`, puoi utilizzarlo in un lavoro di trascrizione.

```
aws transcribe list-language-models
```

Output:

```
{
  "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"
    }
  }
]
}

```

Per ulteriori informazioni, consulta [Modelli linguistici personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per API i dettagli, vedere [ListLanguageModels](#) in AWS CLI Command Reference.

## list-medical-transcription-jobs

Il seguente esempio di codice mostra come utilizzare `list-medical-transcription-jobs`.

### AWS CLI

Elencare i processi di trascrizione medica

L'esempio seguente elenca i lavori di trascrizione medica associati al tuo AWS account e alla tua regione. Per ottenere ulteriori informazioni su un particolare processo di trascrizione, copiate il valore di un `MedicalTranscriptionJobName` parametro nell'output della trascrizione e specificate quel valore per l'opzione del comando. `MedicalTranscriptionJobName get-medical-transcription-job` Per visualizzare altri



lavori di trascrizione, copiate il valore del NextToken parametro, eseguite nuovamente il `list-medical-transcription-jobs` comando e specificate quel valore nell'opzione. `--next-token`

```
aws transcribe list-medical-transcription-jobs
```

Output:

```
{
  "NextToken": "3/PblzkiGhzjER3KHuQt2fmbPLF7cDYafjFMEoGn440N/
gsuUSTIkGyanvRE6WMXFd/ZTEc2EZj+P9eii/
z102FDYli6RLI0WoRX4RwMisVrh9G0Kie0Y8ikBCdtqLZB10Wa9McC+eb0l
+LaDtZPC4u6ttoHLRL1EfzqstHXSgapXg3tEBtm9piIaPB6MOM5BB6t86+qtmocTR/
qrteHZBBudhTfbCwhsxaqujHiiUvFdm3BQbKKWIW06yV9b+4f38oD2lVIan
+vfUs3gBYA15VTDmXXzQPBQ0HPjtwmFI+IWX15nSUjWuN3TUylHgPWzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSenzq49Q9xHgHdT2yBADFmvFK7myZBsJ75+2vQZ0SVpWUPy3WT/32zFAcoEL
+mFYfUjtTZ8n/jq7aQEjQ42A
+X/7K6Jg0cdVPtEg8P1Dr5kgYYG3q30mYXX37U3FZuJmnTI63VtIXsNn0U5eGoY0btpk00Nq9UkzgSJxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSJfUwSAqcFdLIWZdp8qIREMQIBWY7BLwSdyqsQo2vRrd53hm5aWM7SVf6pPq6X/
IXR5+1eU00D8/coaTT4ES2DerbV6RkV4o0VT1d0SdVX/
MmtkNG8nYj8PqU07w7988quh1ZP6D80veJS1q73tUUR9MjnGernW2tAnvnLNhdefBcD
+sZVfYq3iBMFY7wTy1P1G6NqW9GrYDY0X3tTPW1D7phpbVSYKrh/
PdYrps5UxnsGoA1b7L/FfAXDfUoGrGUB4N3JsPYXX9D++g+6gV1qBBs/
Wff934aKqfD6UTggm/zV3GA0WiBpfvAZRvEb924i6yGHYMC7y5401ZAwSBupmI
+FFd13CaP04kN1vJlth6aM5vUPXg4BpyUhtbRhWD/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"
  }
]
```

```
}
```

Per ulteriori informazioni, consulta <https://docs.aws.amazon.com/transcribe/latest/dg/batch-med-transcription.html> nella Amazon Transcribe Developer Guide.

- Per i API dettagli, [ListMedicalTranscriptionJobs AWS CLI](#) consulta Command Reference.

## list-medical-vocabularies

Il seguente esempio di codice mostra come utilizzare `list-medical-vocabularies`.

### AWS CLI

Per elencare i vocabolari medici personalizzati

L'`list-medical-vocabularies` esempio seguente elenca i vocabolari medici personalizzati associati al tuo account e alla tua regione. AWS Per ottenere ulteriori informazioni su un particolare processo di trascrizione, copiate il valore di un `MedicalTranscriptionJobName` parametro nell'output della trascrizione e specificate quel valore per l'opzione del comando. `MedicalTranscriptionJobName get-medical-transcription-job` Per visualizzare altri lavori di trascrizione, copiate il valore del `NextToken` parametro, eseguite nuovamente il `list-medical-transcription-jobs` comando e specificate quel valore nell'opzione. `--next-token`

```
aws transcribe list-medical-vocabularies
```

Output:

```
{
  "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": "2020-09-19T23:59:04.349000+00:00",
```

```
        "VocabularyState": "READY"
      }
    ]
  }
```

Per ulteriori informazioni, consulta [Vocabolari medici personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per API i dettagli, consultate AWS CLI Command [ListMedicalVocabularies](#) Reference.

## list-transcription-jobs

Il seguente esempio di codice mostra come utilizzare `list-transcription-jobs`.

### AWS CLI

Elencare i processi di trascrizione

L'`list-transcription-jobs` esempio seguente elenca i lavori di trascrizione associati al tuo AWS account e alla tua regione.

```
aws transcribe list-transcription-jobs
```

Output:

```
{
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Getting Started \(AWS Command Line Interface\)](#) nella Amazon Transcribe Developer Guide.

- Per API i dettagli, consulta AWS CLI Command [ListTranscriptionJobs](#)Reference.

## list-vocabularies

Il seguente esempio di codice mostra come utilizzare `list-vocabularies`.

## AWS CLI

### Elenco dei vocabolari personalizzati

L'`list-vocabularies` seguente elenca i vocabolari personalizzati associati all' AWS account e alla regione.

```
aws transcribe list-vocabularies
```

### Output:

```
{
  "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"
    }
  ]
}
```

```
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Vocabolari personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per API i dettagli, vedere [ListVocabularies](#) in AWS CLI Command Reference.

## list-vocabulary-filters

Il seguente esempio di codice mostra come utilizzare `list-vocabulary-filters`.

### AWS CLI

Per elencare i filtri del vocabolario

L'`list-vocabulary-filters` esempio seguente elenca i filtri del vocabolario associati al tuo account e alla tua regione. AWS

```
aws transcribe list-vocabulary-filters
```

Output:

```
{  
  "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"
    }
]
}

```

Per ulteriori informazioni, consulta [Filtrare le parole indesiderate](#) nella Amazon Transcribe Developer Guide.

- Per API i dettagli, consulta Command [ListVocabularyFilters](#)Reference AWS CLI .

## start-medical-transcription-job

Il seguente esempio di codice mostra come utilizzare `start-medical-transcription-job`.

### AWS CLI

Esempio 1: trascrivere una dettatura medica archiviata come file audio

L'esempio `start-medical-transcription-job` seguente trascrive un file audio. Specifica la posizione dell'output della trascrizione nel parametro. `OutputBucketName`

```

aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfile.json

```

Contenuto di `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"
  }
}

```



```
}
```

### Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta la [panoramica della trascrizione in batch](#) nella Guida per gli sviluppatori di Amazon Transcribe.

### Esempio 2: trascrivere un dialogo medico-paziente archiviato come file audio

L'esempio `start-medical-transcription-job` seguente trascrive un file audio contenente un dialogo medico-paziente. Si specifica la posizione dell'output della trascrizione nel `OutputBucketName` parametro.

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file://mysecondfile.json
```

Contenuto di `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"
  }
}
```

```
}
```

### Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta la [panoramica della trascrizione in batch](#) nella Guida per gli sviluppatori di Amazon Transcribe.

### Esempio 3: trascrivere un file audio multicanale di un dialogo medico-paziente

L'esempio `start-medical-transcription-job` seguente trascrive l'audio da ogni canale del file audio e unisce le trascrizioni separate di ciascun canale in un unico batch di trascrizione. Specifica la posizione dell'output della trascrizione nel parametro `OutputBucketName`

```
aws transcribe start-medical-transcription-job \
  --cli-input-json file:///mythirdfile.json
```

Contenuto di `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
    }
  }
}

```

### Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Identificazione del canale](#) nella Guida per gli sviluppatori di Amazon Transcribe.

Esempio 4: trascrivere un file audio di un dialogo tra medico e paziente e identificare le persone che parlano nell'output della trascrizione

L'esempio `start-medical-transcription-job` seguente trascrive un file audio ed etichetta il discorso di ciascuna persona che parla nell'output di trascrizione. Specifica la posizione dell'output della trascrizione nel parametro. `OutputBucketName`

```

aws transcribe start-medical-transcription-job \
  --cli-input-json file://myfourthfile.json

```

## Contenuto di 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
  }
}
```

## Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Identificazione delle persone che parlano](#) nella Guida per gli sviluppatori di Amazon Transcribe.

## Esempio 5: trascrivere una conversazione medica archiviata come file audio con un massimo di due alternative di trascrizione

L'esempio `start-medical-transcription-job` seguente crea fino a due trascrizioni alternative da un singolo file audio. A ogni trascrizione è associato un livello di affidabilità. Per impostazione predefinita, Amazon Transcribe restituisce la trascrizione con il livello di affidabilità più elevato. Puoi specificare che Amazon Transcribe restituisca trascrizioni aggiuntive con livelli di affidabilità inferiori. Specifica la posizione dell'output della trascrizione nel parametro `OutputBucketName`

```
aws transcribe start-medical-transcription-job \  
  --cli-input-json file://myfifthfile.json
```

Contenuto di `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  
  }  
}
```

Output:

```
{  
  "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"
  }
}

```

Per ulteriori informazioni, consulta [Trascrizioni alternative](#) nella Guida per gli sviluppatori di Amazon Transcribe.

Esempio 6: trascrivere un file audio di una dettatura medica con un massimo di due trascrizioni alternative

L'esempio `start-medical-transcription-job` seguente trascrive un file audio e utilizza un filtro del vocabolario per mascherare eventuali parole indesiderate. Si specifica la posizione dell'output della trascrizione nel parametro. `OutputBucketName`

```

aws transcribe start-medical-transcription-job \
  --cli-input-json file://mysixthfile.json

```

Contenuto di `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
  }
}

```

**Output:**

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Trascrizioni alternative](#) nella Guida per gli sviluppatori di Amazon Transcribe.

Esempio 7: trascrivere un file audio di una dettatura medica con maggiore precisione utilizzando un vocabolario personalizzato

L'esempio `start-medical-transcription-job` seguente trascrive un file audio e utilizza un vocabolario medico personalizzato creato in precedenza per aumentare la precisione della trascrizione. Specifica la posizione dell'output della trascrizione nel parametro.

`OutputBucketName`

```
aws transcribe start-transcription-job \
  --cli-input-json file://myseventhfile.json
```

Contenuto di `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"
}
}
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta [Vocabolari medici personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per API i dettagli, vedere [StartMedicalTranscriptionJob](#) in AWS CLI Command Reference.

## start-transcription-job

Il seguente esempio di codice mostra come utilizzare `start-transcription-job`.

### AWS CLI

Esempio 1: trascrivere un file audio



L'esempio `start-transcription-job` seguente trascrive il file audio.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://myfile.json
```

Contenuto di `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"  
  }  
}
```

Per ulteriori informazioni, consulta [Getting Started \(AWS Command Line Interface\)](#) nella Amazon Transcribe Developer Guide.

Esempio 2: trascrivere un file audio multicanale

L'esempio `start-transcription-job` seguente trascrive il file audio multicanale.

```
aws transcribe start-transcription-job \  
  --cli-input-json file://mysecondfile.json
```

Contenuto di `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  
  }  
}
```

Output:

```
{
  "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
    }
  }
}
```

Per ulteriori informazioni, consulta [Trascrizione dell'audio multicanale](#) nella Guida per gli sviluppatori di Amazon Transcribe.

Esempio 3: trascrivere un file audio e identificare le diverse persone che parlano

L'esempio `start-transcription-job` seguente trascrive il file audio e identifica le persone che parlano nell'output di trascrizione.

```
aws transcribe start-transcription-job \
  --cli-input-json file://mythirdfile.json
```

Contenuto di `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
  }
}
```

**Output:**

```
{
  "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
    }
  }
}
```

Per ulteriori informazioni, consulta [Identificazione delle persone che parlano](#) nella Guida per gli sviluppatori di Amazon Transcribe.

Esempio 4: trascrivere un file audio e mascherare eventuali parole indesiderate nell'output della trascrizione

L'esempio `start-transcription-job` seguente trascrive il file audio e utilizza un filtro del vocabolario creato in precedenza per mascherare eventuali parole indesiderate.

```
aws transcribe start-transcription-job \
  --cli-input-json file://myfourthfile.json
```

Contenuto di `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"
  }
}

```

### Output:

```

{
  "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"
    }
  }
}

```

Per ulteriori informazioni, consulta [Filtraggio delle trascrizioni](#) nella Guida per gli sviluppatori di Amazon Transcribe.

Esempio 5: trascrivere un file audio e rimuovere eventuali parole indesiderate nell'output della trascrizione

L'esempio `start-transcription-job` seguente trascrive il file audio e utilizza un filtro del vocabolario creato in precedenza per mascherare eventuali parole indesiderate.

```

aws transcribe start-transcription-job \
  --cli-input-json file://myfifthfile.json

```

Contenuto di `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"
  }
}

```

### Output:

```

{
  "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"
    }
  }
}

```

Per ulteriori informazioni, consulta [Filtraggio delle trascrizioni](#) nella Guida per gli sviluppatori di Amazon Transcribe.

Esempio 6: trascrivere un file audio con maggiore precisione utilizzando un vocabolario personalizzato

L'esempio `start-transcription-job` seguente trascrive il file audio e utilizza un filtro del vocabolario creato in precedenza per mascherare eventuali parole indesiderate.

```

aws transcribe start-transcription-job \
  --cli-input-json file://mysixthfile.json

```

Contenuto di `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"
  }
}
```

Output:

```
{
  "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"
    }
  }
}
```

Per ulteriori informazioni, consulta [Filtraggio delle trascrizioni](#) nella Guida per gli sviluppatori di Amazon Transcribe.

Esempio 7: identificare la lingua di un file audio e trascriverla

L'esempio `start-transcription-job` seguente trascrive il file audio e utilizza un filtro del vocabolario creato in precedenza per mascherare eventuali parole indesiderate.

```
aws transcribe start-transcription-job \
  --cli-input-json file://myseventhfile.json
```

## Contenuto di 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"
  }
}
```

## Output:

```
{
  "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
  }
}
```

Per ulteriori informazioni, consulta [Identificazione della lingua](#) nella Guida per gli sviluppatori di Amazon Transcribe.

## Esempio 8: trascrivere un file audio con informazioni di identificazione personale nascoste

L'esempio `start-transcription-job` seguente trascrive il file audio e nasconde le informazioni di identificazione personale nell'output di trascrizione.

```
aws transcribe start-transcription-job \
  --cli-input-json file://myeighthfile.json
```

## Contenuto di 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"
    }
  }
}

```

### Output:

```

{
  "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"
    }
  }
}

```

Per ulteriori informazioni, consulta [Redazione dei contenuti automatici](#) nella Guida per gli sviluppatori di Amazon Transcribe.

Esempio 9: generare una trascrizione con informazioni di identificazione personale (PII) redatta e una trascrizione non redatta

L'esempio `start-transcription-job` seguente genera due trascrizioni del file audio, una con le informazioni di identificazione personale redatte e l'altra senza alcuna redazione.

```

aws transcribe start-transcription-job \
  --cli-input-json file://myninthfile.json

```

Contenuto di `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"
  }
}
```

Output:

```
{
  "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"
    }
  }
}
```

Per ulteriori informazioni, consulta [Redazione dei contenuti automatici](#) nella Guida per gli sviluppatori di Amazon Transcribe.

Esempio 10: utilizzare un modello linguistico personalizzato creato in precedenza per trascrivere un file audio.

L'esempio `start-transcription-job` seguente trascrive il file audio con un modello linguistico personalizzato creato in precedenza.

```
aws transcribe start-transcription-job \
```

```
--cli-input-json file://mytenthfile.json
```

Contenuto di 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"
  }
}
```

Output:

```
{
  "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"
    }
  }
}
```

Per ulteriori informazioni, consulta [Modelli linguistici personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per i API dettagli, consulta Command Reference. [StartTranscriptionJob](#)AWS CLI

## update-medical-vocabulary

Il seguente esempio di codice mostra come utilizzare `update-medical-vocabulary`.

## AWS CLI

Per aggiornare un vocabolario medico personalizzato con nuovi termini.

L'`update-medical-vocabulary` esempio seguente sostituisce i termini utilizzati in un vocabolario medico personalizzato con quelli nuovi. Prerequisito: per sostituire i termini in un vocabolario medico personalizzato, è necessario un file con nuovi termini.

```
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
```

Output:

```
{  
  "VocabularyName": "medical-custom-vocabulary",  
  "LanguageCode": "en-US",  
  "VocabularyState": "PENDING"  
}
```

Per ulteriori informazioni, consulta [Vocabolari medici personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per i API dettagli, vedere [UpdateMedicalVocabulary](#) in Command Reference.AWS CLI

## update-vocabulary-filter

Il seguente esempio di codice mostra come utilizzare `update-vocabulary-filter`.

## AWS CLI

Per sostituire le parole in un filtro del vocabolario

L'`update-vocabulary-filter` esempio seguente sostituisce le parole in un filtro del vocabolario con altre nuove. Prerequisito: per aggiornare un filtro del vocabolario con le nuove parole, è necessario che tali parole siano salvate come file di testo.

```
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 \  
  --language-code language
```

```
--vocabulary-filter-name vocabulary-filter-name
```

Output:

```
{
  "VocabularyFilterName": "vocabulary-filter-name",
  "LanguageCode": "language-code",
  "LastModifiedTime": "2020-09-23T18:40:35.139000+00:00"
}
```

Per ulteriori informazioni, consulta [Filtrare le parole indesiderate](#) nella Amazon Transcribe Developer Guide.

- Per API i dettagli, consulta Command [UpdateVocabularyFilter](#)Reference AWS CLI .

## update-vocabulary

Il seguente esempio di codice mostra come utilizzare `update-vocabulary`.

### AWS CLI

Aggiornamento di un vocabolario personalizzato con nuovi termini.

L'esempio `update-vocabulary` seguente sovrascrive i termini utilizzati per creare un vocabolario personalizzato con quelli nuovi forniti dall'utente. Prerequisito: per sostituire i termini in un vocabolario personalizzato, è necessario disporre di un file con nuovi termini.

```
aws transcribe update-vocabulary \
  --vocabulary-file-uri s3://DOC-EXAMPLE-BUCKET/Amazon-S3-Prefix/custom-
vocabulary.txt \
  --vocabulary-name custom-vocabulary \
  --language-code language-code
```

Output:

```
{
  "VocabularyName": "custom-vocabulary",
  "LanguageCode": "language",
  "VocabularyState": "PENDING"
}
```

Per ulteriori informazioni, consulta [Vocabolari personalizzati](#) nella Guida per gli sviluppatori di Amazon Transcribe.

- Per API i dettagli, vedere [UpdateVocabulary](#) in AWS CLI Command Reference.

## Esempi di Amazon Translate con AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon Translate. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **import-terminology**

Il seguente esempio di codice mostra come utilizzare `import-terminology`.

AWS CLI

Per importare una terminologia personalizzata da un file

L'`import-terminology` esempio seguente crea una terminologia richiamata `MyTestTerminology` dal `test-terminology.csv` file:

```
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
```

Contenuto di `test-terminology.csv`.

en, fr, es, zh Ciao mondo! , Bonjour tout le monde! , Hola Mundo! ,???? Amazon, Amazon, Amazon, Amazon

Output:

```
{
  "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"
  }
}
```

- Per API i dettagli, consulta [ImportTerminology AWS CLI](#) Command Reference.

## Trusted Advisor esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Trusted Advisor.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

## Azioni

### **get-organization-recommendation**

Il seguente esempio di codice mostra come utilizzare `get-organization-recommendation`.

#### AWS CLI

Per ottenere una raccomandazione sull'organizzazione

L'`get-organization-recommendation` seguente ottiene una raccomandazione di un'organizzazione in base al relativo identificatore.

```
aws trustedadvisor get-organization-recommendation \
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5
```

#### Output:

```
{
  "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"
  }
}
```

```
}
```

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

- Per API i dettagli, vedere [GetOrganizationRecommendation](#) in AWS CLI Command Reference.

## get-recommendation

Il seguente esempio di codice mostra come utilizzare `get-recommendation`.

### AWS CLI

Per ricevere una raccomandazione

L'`get-recommendation` esempio seguente ottiene una raccomandazione in base al relativo identificatore.

```
aws trustedadvisor get-recommendation \
  --recommendation-
  identifier arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
  bbb7-491a-833b-5773e9589578
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

- Per API i dettagli, vedere [GetRecommendation](#) in AWS CLI Command Reference.

## list-checks

Il seguente esempio di codice mostra come utilizzare `list-checks`.

### AWS CLI

Per elencare i controlli di Trusted Advisor

L'`list-checks` esempio seguente elenca tutti i controlli Trusted Advisor.

```
aws trustedadvisor list-checks
```

Output:

```
{
  "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"
}

```

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

- Per API i dettagli, vedere [ListChecks](#) in AWS CLI Command Reference.

## list-organization-recommendation-accounts

Il seguente esempio di codice mostra come utilizzare `list-organization-recommendation-accounts`.

### AWS CLI

Per elencare gli account di raccomandazione dell'organizzazione

L'`list-organization-recommendation-accounts` esempio seguente elenca tutti i riepiloghi dei consigli relativi agli account relativi a un'organizzazione in base al relativo identificatore.

```

aws trustedadvisor list-organization-recommendation-accounts \
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5

```

Output:

```

{
  "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"
}
```

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

- Per API i dettagli, vedere [ListOrganizationRecommendationAccounts](#) in AWS CLI Command Reference.

## list-organization-recommendation-resources

Il seguente esempio di codice mostra come utilizzare `list-organization-recommendation-resources`.

### AWS CLI

Per elencare le risorse di raccomandazione dell'organizzazione

L'`list-organization-recommendation-resources` esempio seguente elenca tutte le risorse per una raccomandazione organizzativa in base al relativo identificatore.

```
aws trustedadvisor list-organization-recommendation-resources \
  --organization-recommendation-identifier arn:aws:trustedadvisor::organization-recommendation/5a694939-2e54-45a2-ae72-730598fa89d0
```

Output:

```
{
  "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"
}

```

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

- Per API i dettagli, vedere [ListOrganizationRecommendationResources](#) in AWS CLI Command Reference.

## list-organization-recommendations

Il seguente esempio di codice mostra come utilizzare `list-organization-recommendations`.

### AWS CLI

Esempio 1: per elencare i consigli sull'organizzazione

L'`list-organization-recommendations` seguente elenca tutti i consigli sull'organizzazione e non include un filtro.

```
aws trustedadvisor list-organization-recommendations
```

Output:

```

{
  "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"
```



```
}
```

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

Esempio 2: per elencare i consigli sull'organizzazione con un filtro

L'`list-organization-recommendation` seguente filtra e restituisce al massimo una raccomandazione aziendale che fa parte del pilastro «sicurezza».

```
aws trustedadvisor list-organization-recommendations \  
  --pillar security \  
  --max-items 100
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

Esempio 3: Per elencare i consigli sull'organizzazione con un token di impaginazione

L'`list-organization-recommendations` seguente utilizza il "nextToken" restituito da una richiesta precedente per recuperare la pagina successiva di consigli sull'organizzazione.

```
aws trustedadvisor list-organization-recommendations \  
  --pillar security \  
  --max-items 100 \  
  --starting-token <next-token>
```

Output:

```
{  
  "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"  
  }]  
}
```

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

- Per API i dettagli, vedere [ListOrganizationRecommendations](#) in AWS CLI Command Reference.

## list-recommendation-resources

Il seguente esempio di codice mostra come utilizzare `list-recommendation-resources`.

### AWS CLI

Per elencare le risorse di raccomandazione

L'`list-recommendation-resources` esempio seguente elenca tutte le risorse per un consiglio in base al relativo identificatore.

```
aws trustedadvisor list-recommendation-resources \
  --recommendation-
  identifier arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
  bbb7-491a-833b-5773e9589578
```

Output:

```
{
  "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": "123.120000000000002",
        "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/
e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcbce4b9e4fefcec9eb63e",
        "id":
        "e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcbce4b9e4fefcec9eb63e",
        "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"
}

```

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

- Per API i dettagli, vedere [ListRecommendationResources](#) in AWS CLI Command Reference.

## list-recommendations

Il seguente esempio di codice mostra come utilizzare `list-recommendations`.

### AWS CLI

Esempio 1: per elencare i consigli

L'`list-recommendations` esempio seguente elenca tutti i consigli e non include un filtro.

```
aws trustedadvisor list-recommendations
```

Output:

```

{
  "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"
}

```

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

Esempio 2: per elencare i consigli con un filtro

L'`list-recommendations` seguente elenca i consigli e include un filtro.

```

aws trustedadvisor list-recommendations \
  --aws-service iam \
  --max-items 100

```

Output:

```

{
  "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"
  ]],
  "nextToken": "REDACTED"
}

```

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

Esempio 3: Per elencare i consigli con un token di impaginazione

L'`list-recommendations` seguente utilizza il "nextToken" restituito da una richiesta precedente per recuperare la pagina successiva di Consigli filtrati.

```

aws trustedadvisor list-recommendations \
  --aws-service rds \
  --max-items 100 \
  --starting-token <next-token>

```

Output:

```

{
  "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"
  ]]
}

```

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

- Per API i dettagli, vedere [ListRecommendations](#) in AWS CLI Command Reference.

## update-organization-recommendation-lifecycle

Il seguente esempio di codice mostra come utilizzare `update-organization-recommendation-lifecycle`.

AWS CLI

Per aggiornare il ciclo di vita dei consigli di un'organizzazione

L'`update-organization-recommendation-lifecycle` esempio seguente aggiorna il ciclo di vita di una raccomandazione dell'organizzazione in base al relativo identificatore.

```

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

```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

- Per API i dettagli, vedere [UpdateOrganizationRecommendationLifecycle](#) in AWS CLI Command Reference.

## update-recommendation-lifecycle

Il seguente esempio di codice mostra come utilizzare `update-recommendation-lifecycle`.

### AWS CLI

Per aggiornare il ciclo di vita di una raccomandazione

L'esempio seguente aggiorna il ciclo di vita di una raccomandazione in base al relativo identificatore.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la Guida [introduttiva al Trusted Advisor API](#) nella AWS Trusted Advisor User Guide.

- Per API i dettagli, vedere [UpdateRecommendationLifecycle](#) in AWS CLI Command Reference.

## Esempi di autorizzazioni verificate utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with Verified Permissions.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### **create-identity-source**

Il seguente esempio di codice mostra come utilizzare `create-identity-source`.

#### AWS CLI

Per creare una fonte di identità

L'esempio seguente crea un'origine di identità che consente di fare riferimento alle identità archiviate nel pool di utenti Amazon Cognito specificato. Queste identità sono disponibili in Autorizzazioni verificate come entità di tipo `User`

```
aws verifiedpermissions create-identity-source \  
  --configuration file://config.txt \  
  --principal-entity-type "User" \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Contenuto di `config.txt`.

```
{  
  "cognitoUserPoolConfiguration": {  
    "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-  
west-2_1a2b3c4d5",  
    "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"]  
  }  
}
```

Output:

```
{  
  "createdDate": "2023-05-19T20:30:28.214829+00:00",  
  "identitySourceId": "ISEXAMPLEabcdefg111111",  
  "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",  
  "policyStoreId": "PSEXAMPLEabcdefg111111"  
}
```

Per ulteriori informazioni sulle fonti di identità, consulta [Using Amazon Verified Permissions with identity provider](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateIdentitySource](#)Reference.

## create-policy-store

Il seguente esempio di codice mostra come utilizzare `create-policy-store`.

### AWS CLI

Per creare un archivio delle politiche

L'esempio seguente crea un archivio delle politiche nella AWS regione corrente.

```
aws verifiedpermissions create-policy-store \
  --validation-settings "mode=STRICT"
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni sugli archivi delle politiche, consulta gli archivi [delle politiche di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreatePolicyStore](#)Reference.

## create-policy-template

Il seguente esempio di codice mostra come utilizzare `create-policy-template`.

### AWS CLI

Esempio 1: creare un modello di policy

L'`create-policy-template` seguente crea un modello di policy con un'istruzione che contiene un segnaposto per il principale.

```
aws verifiedpermissions create-policy-template \  
  --definition file://template1.txt \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Contenuto del file: `template1.txt`

```
permit(  
  principal in ?principal,  
  action == Action::"view",  
  resource == Photo::"VacationPhoto94.jpg"  
);
```

Output:

```
{  
  "createdDate": "2023-06-12T20:47:42.804511+00:00",  
  "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",  
  "policyStoreId": "PSEXAMPLEEabcdefg111111",  
  "policyTemplateId": "PTEXAMPLEEabcdefg111111"  
}
```

Per ulteriori informazioni sui modelli di policy, consulta i modelli di [policy di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreatePolicyTemplateReference](#).

## **create-policy**

Il seguente esempio di codice mostra come utilizzare `create-policy`.

### AWS CLI

Esempio 1: creare una politica statica

L'`create-policy` seguente crea una politica statica con un ambito di policy che specifica sia un principale che una risorsa.

```
aws verifiedpermissions create-policy \  
  --definition file://template1.txt \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

```
--definition file://definition1.txt \  
--policy-store-id PSEXAMPLEabcdefgh111111
```

Contenuto del filedefinition1.txt:

```
{  
  "static": {  
    "description": "Grant everyone of janeFriends UserGroup access to the  
vacationFolder Album",  
    "statement": "permit(principal in UserGroup::\"janeFriends\", action,  
resource in Album::\"vacationFolder\" );"  
  }  
}
```

Output:

```
{  
  "createdDate": "2023-06-12T20:33:37.382907+00:00",  
  "lastUpdatedDate": "2023-06-12T20:33:37.382907+00:00",  
  "policyId": "SPEXAMPLEabcdefgh111111",  
  "policyStoreId": "PSEXAMPLEabcdefgh111111",  
  "policyType": "STATIC",  
  "principal": {  
    "entityId": "janeFriends",  
    "entityType": "UserGroup"  
  },  
  "resource": {  
    "entityId": "vacationFolder",  
    "entityType": "Album"  
  }  
}
```

Esempio 2: creare una politica statica che garantisca l'accesso a una risorsa a tutti

L'create-policyesempio seguente crea una politica statica con un ambito di policy che specifica solo una risorsa.

```
aws verifiedpermissions create-policy \  
--definition file://definition2.txt \  
--policy-store-id PSEXAMPLEabcdefgh111111
```

Contenuto del filedefinition2.txt:

```
{
  "static": {
    "description": "Grant everyone access to the publicFolder Album",
    "statement": "permit(principal, action, resource in Album:\""publicFolder
  \");"
  }
}
```

Output:

```
{
  "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"
  }
}
```

Esempio 3: creare una politica collegata al modello associata al modello specificato

L'`create-policy` seguente crea una politica collegata al modello utilizzando il modello di policy specificato e associa il principale specificato da utilizzare alla nuova politica collegata al modello.

```
aws verifiedpermissions create-policy \
  --definition file://definition.txt \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Contenuto di `definition.txt`.

```
{
  "templateLinked": {
    "policyTemplateId": "PTEXAMPLEEabcdefg111111",
    "principal": {
      "entityType": "User",
      "entityId": "alice"
    }
  }
}
```

```
}  
}
```

Output:

```
{  
  "createdDate": "2023-06-12T20:49:51.490211+00:00",  
  "lastUpdatedDate": "2023-06-12T20:49:51.490211+00:00",  
  "policyId": "TPEXAMPLEabcdefg111111",  
  "policyStoreId": "PSEXAMPLEabcdefg111111",  
  "policyType": "TEMPLATE_LINKED",  
  "principal": {  
    "entityId": "alice",  
    "entityType": "User"  
  },  
  "resource": {  
    "entityId": "VacationPhoto94.jpg",  
    "entityType": "Photo"  
  }  
}
```

Per ulteriori informazioni sulle politiche, consulta le [politiche di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreatePolicy](#) Reference.

## delete-identity-source

Il seguente esempio di codice mostra come utilizzare `delete-identity-source`.

AWS CLI

Per eliminare una fonte di identità

L'`delete-identity-source` esempio seguente elimina l'origine dell'identità con l'ID specificato.

```
aws verifiedpermissions delete-identity-source \  
  --identity-source-id ISEXAMPLEabcdefg111111 \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Questo comando non produce alcun output.



Per ulteriori informazioni sulle fonti di identità, consulta [Using Amazon Verified Permissions with identity provider](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteIdentitySource](#)Reference.

## delete-policy-store

Il seguente esempio di codice mostra come utilizzare `delete-policy-store`.

### AWS CLI

Per eliminare un archivio delle politiche

L'`delete-policy-store`esempio seguente elimina il policy store con l'ID specificato.

```
aws verifiedpermissions delete-policy-store \  
  --policy-store-id PEXAMPLEabcdefg111111
```

Questo comando non produce alcun output.

Per ulteriori informazioni sugli archivi delle politiche, consulta gli archivi [delle politiche di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeletePolicyStore](#)Reference.

## delete-policy-template

Il seguente esempio di codice mostra come utilizzare `delete-policy-template`.

### AWS CLI

Per eliminare un modello di policy

L'`delete-policy-template`esempio seguente elimina il modello di policy con l'ID specificato.

```
aws verifiedpermissions delete-policy \  
  --policy-template-id PEXAMPLEabcdefg111111 \  
  --policy-store-id PEXAMPLEabcdefg111111
```

Questo comando non produce alcun output.

Per ulteriori informazioni sui modelli di policy, consulta i modelli di [policy di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeletePolicyTemplateReference](#).

## delete-policy

Il seguente esempio di codice mostra come utilizzare `delete-policy`.

### AWS CLI

Per eliminare una policy statica o collegata a un modello

L'`delete-policy` esempio seguente elimina la politica con l'ID specificato.

```
aws verifiedpermissions delete-policy \  
  --policy-id SPEXAMPLEEabcdefg111111 \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Questo comando non produce alcun output.

Per ulteriori informazioni sulle politiche, consulta le [politiche di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeletePolicyReference](#).

## get-identity-source

Il seguente esempio di codice mostra come utilizzare `get-identity-source`.

### AWS CLI

Per recuperare i dettagli su una fonte di identità

L'`get-identity-source` esempio seguente visualizza i dettagli dell'origine dell'identità con l'ID specificato.

```
aws verifiedpermissions get-identity-source \  
  --identity-source ISEXAMPLEEabcdefg111111 \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

**Output:**

```
{
  "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": "ISEXAMPLEabcdefgh111111",
  "lastUpdatedDate": "2023-06-12T22:27:49.150035+00:00",
  "policyStoreId": "PSEXAMPLEabcdefgh111111",
  "principalEntityType": "User"
}
```

Per ulteriori informazioni sulle fonti di identità, consulta [Using Amazon Verified Permissions with identity provider](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetIdentitySource](#)Reference.

**get-policy-store**

Il seguente esempio di codice mostra come utilizzare `get-policy-store`.

**AWS CLI**

Per recuperare i dettagli su un archivio di politiche

L'`get-policy-store`esempio seguente visualizza i dettagli del policy store con l'ID specificato.

```
aws verifiedpermissions get-policy-store \
  --policy-store-id PSEXAMPLEabcdefgh111111
```

**Output:**

```
{
  "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
  PSEXAMPLEabcdefgh111111",
```

```
"createdDate": "2023-06-05T20:16:46.225598+00:00",
"lastUpdatedDate": "2023-06-08T20:40:23.173691+00:00",
"policyStoreId": "PSEXAMPLEEabcdefg111111",
"validationSettings": { "mode": "OFF" }
}
```

Per ulteriori informazioni sugli archivi delle politiche, consulta gli archivi [delle politiche di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetPolicyStore](#)Reference.

## get-policy-template

Il seguente esempio di codice mostra come utilizzare `get-policy-template`.

### AWS CLI

Per recuperare i dettagli su un modello di policy

L'`get-policy-template` esempio seguente visualizza i dettagli del modello di policy con l'ID specificato.

```
aws verifiedpermissions get-policy-template \
  --policy-template-id PTEXAMPLEEabcdefg111111 \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Output:

```
{
  "createdDate": "2023-06-12T20:47:42.804511+00:00",
  "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
  "policyStoreId": "PSEXAMPLEEabcdefg111111",
  "policyTemplateId": "PTEXAMPLEEabcdefg111111",
  "statement": "permit(\n  principal in ?principal,\n  action == Action::\n  \"view\", \n  resource == Photo::\"VacationPhoto94.jpg\" \n);"
}
```

Per ulteriori informazioni sui modelli di policy, consulta i modelli di [policy di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetPolicyTemplate](#)Reference.

## get-policy

Il seguente esempio di codice mostra come utilizzare `get-policy`.

### AWS CLI

Per recuperare i dettagli su una politica

L'`get-policy`esempio seguente visualizza i dettagli della politica con l'ID specificato.

```
aws verifiedpermissions get-policy \  
  --policy-id PSEXAMPLEabcdefg111111 \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni sulle politiche, consulta le [politiche di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetPolicyReference](#).

## get-schema

Il seguente esempio di codice mostra come utilizzare `get-schema`.

### AWS CLI

Per recuperare lo schema in un archivio di politiche

L'`get-schema` esempio seguente visualizza i dettagli dello schema nel policy store specificato.

```
aws verifiedpermissions get-schema \
  --policy-store-id PSEXAMPLEabcdefg111111
```

Output:

```
{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "schema": "{\\\"MySampleNamespace\\\":{\\\"entityTypes\\\":{\\\"Employee\\\":{\\\"shape\\\":{\\\"attributes\\\":{\\\"jobLevel\\\":{\\\"type\\\":\\\"Long\\\"},\\\"name\\\":{\\\"type\\\":\\\"String\\\"}},\\\"type\\\":\\\"Record\\\"}}},\\\"actions\\\":{\\\"remoteAccess\\\":{\\\"appliesTo\\\":{\\\"principalTypes\\\":[\\\"Employee\\\"]}}}}}",
  "createdDate": "2023-06-14T17:47:13.999885+00:00",
  "lastUpdatedDate": "2023-06-14T17:47:13.999885+00:00"
}
```

Per ulteriori informazioni sullo schema, consulta lo [schema del Policy store](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta [GetSchema AWS CLI](#) Command Reference.

## is-authorized-with-token

Il seguente esempio di codice mostra come utilizzare `is-authorized-with-token`.

### AWS CLI

Esempio 1: richiedere una decisione di autorizzazione per una richiesta dell'utente (consentire)

L'`is-authorized-with-token` esempio seguente richiede una decisione di autorizzazione per un utente autenticato da Amazon Cognito. La richiesta utilizza il token di identità fornito da Cognito anziché il token di accesso. In questo esempio, l'archivio di informazioni specificato è configurato per restituire i principali come entità di tipo. `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"
```

Il policy store contiene una policy con la seguente dichiarazione che accetta le identità dal pool di utenti e dall'ID dell'applicazione Cognito specificati.

```
permit(  
  principal == CognitoUser::"us-east-1_1a2b3c4d5|a1b2c3d4e5f6g7h8i9j0kalbmc",  
  action,  
  resource == Photo::"VacationPhoto94.jpg"  
);
```

Output:

```
{  
  "decision":"Allow",  
  "determiningPolicies":[  
    {  
      "determiningPolicyId":"SPEXAMPLEabcdefg111111"  
    }  
  ],  
  "errors":[]  
}
```

Per ulteriori informazioni sull'utilizzo delle identità di un pool di utenti Cognito, consulta [Using Amazon Verified Permissions con provider di identità](#) nella Amazon Verified Permissions User Guide.

- Per i API dettagli, consulta [IsAuthorizedWithToken](#) Command Reference.AWS CLI

## is-authorized

Il seguente esempio di codice mostra come utilizzare `is-authorized`.

### AWS CLI

Esempio 1: richiedere una decisione di autorizzazione per una richiesta dell'utente (consentire)

L'`is-authorized` seguente richiede una decisione di autorizzazione per un principale di tipo `User` named `Alice`, che desidera eseguire l'`updatePhoto` operazione, su una risorsa di tipo `Photo` denominato `VacationPhoto94.jpg`.

La risposta mostra che la richiesta è consentita da una politica.

```
aws verifiedpermissions is-authorized \  
  --principal entityType=User,entityId=alice \  
  --action actionType=Action,actionId=view \  
  --resource entityType=Photo,entityId=VactionPhoto94.jpg \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Output:

```
{  
  "decision": "ALLOW",  
  "determiningPolicies": [  
    {  
      "policyId": "SPEXAMPLEabcdefg111111"  
    }  
  ],  
  "errors": []  
}
```

Esempio 2: richiedere una decisione di autorizzazione per una richiesta dell'utente (negare)

L'esempio seguente è lo stesso dell'esempio precedente, tranne per il fatto che il principale è `User::"Bob"`. L'archivio delle politiche non contiene alcuna politica che consenta l'accesso a tale utente `Album::"alice_folder"`.

L'output indica che `Deny` era implicito perché l'elenco di `DeterminingPolicies` è vuoto.

```
aws verifiedpermissions create-policy \  
  --definition file://definition2.txt \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Output:

```
{  
  "decision": "DENY",  
  "determiningPolicies": [],
```



```
"errors": []
}
```

Per ulteriori informazioni, consulta la [Amazon Verified Permissions User Guide](#).

- Per API i dettagli, consulta [IsAuthorized AWS CLI Command Reference](#).

## list-identity-sources

Il seguente esempio di codice mostra come utilizzare `list-identity-sources`.

### AWS CLI

Per elencare le fonti di identità disponibili

L'`list-identity-sources` esempio seguente elenca tutte le fonti di identità nell'archivio delle politiche specificato.

```
aws verifiedpermissions list-identity-sources \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni sulle fonti di identità, consulta [Using Amazon Verified Permissions with identity provider](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [ListIdentitySources](#) Reference.

## list-policies

Il seguente esempio di codice mostra come utilizzare `list-policies`.

### AWS CLI

Per elencare le politiche disponibili

L'`list-policies` esempio seguente elenca tutte le politiche nel policy store specificato.

```
aws verifiedpermissions list-policies \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Output:

```
{  
  "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": "SPEXAMPLEabcdefg111111",  
      "policyStoreId": "PSEXAMPLEabcdefg111111",  
      "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": "SPEXAMPLEabcdefg222222",
      "policyStoreId": "PSEXAMPLEabcdefg111111",
      "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"
      }
    }
  ]
}
```

Per ulteriori informazioni sulle politiche, consulta le [politiche di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [ListPolicies](#)Reference.

## list-policy-stores

Il seguente esempio di codice mostra come utilizzare `list-policy-stores`.

### AWS CLI

Per elencare gli archivi di policy disponibili

L'`list-policy-stores` seguente elenca tutti gli archivi delle politiche della AWS regione. Tutti i comandi per le autorizzazioni verificate, ad eccezione `create-policy-store` dei comandi che `list-policy-stores` richiedono la specificazione dell'ID del policy store con cui si desidera lavorare.

```
aws verifiedpermissions list-policy-stores
```

Output:

```
{
  "policyStores": [
    {
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/PSEXAMPLEabcdefg111111",
      "createdDate": "2023-06-05T20:16:46.225598+00:00",
      "policyStoreId": "PSEXAMPLEabcdefg111111"
    },
    {
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/PSEXAMPLEabcdefg222222",
      "createdDate": "2023-06-08T18:09:37.364356+00:00",
      "policyStoreId": "PSEXAMPLEabcdefg222222"
    },
    {
      "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/PSEXAMPLEabcdefg333333",
      "createdDate": "2023-06-08T18:09:46.920600+00:00",
      "policyStoreId": "PSEXAMPLEabcdefg333333"
    }
  ]
}
```

Per ulteriori informazioni sugli archivi delle politiche, consulta gli archivi [delle politiche di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [ListPolicyStoresReference](#).

## list-policy-templates

Il seguente esempio di codice mostra come utilizzare `list-policy-templates`.

### AWS CLI

Per elencare i modelli di policy disponibili

L'`list-policy-templates` esempio seguente elenca tutti i modelli di policy nel policy store specificato.

```
aws verifiedpermissions list-policy-templates \  
  --policy-store-id PEXAMPLEabcdefg111111
```

Output:

```
{  
  "policyTemplates": [  
    {  
      "createdDate": "2023-06-12T20:47:42.804511+00:00",  
      "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",  
      "policyStoreId": "PEXAMPLEabcdefg111111",  
      "policyTemplateId": "PTEXAMPLEabcdefg111111"  
    }  
  ]  
}
```

Per ulteriori informazioni sui modelli di policy, consulta i modelli di [policy di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [ListPolicyTemplatesReference](#).

## put-schema

Il seguente esempio di codice mostra come utilizzare `put-schema`.

### AWS CLI

Per salvare uno schema in un archivio di politiche

L'put-schemaesempio seguente crea o sostituisce lo schema nel policy store specificato.

Il cedarJson parametro nel file di input accetta una rappresentazione in formato stringa di un JSON oggetto. Contiene virgolette incorporate («) all'interno della coppia di virgolette più esterna. A tale scopo è necessario JSON convertire il in stringa facendo precedere tutte le virgolette incorporate da una barra rovesciata (") e combinando tutte le righe in un'unica riga di testo senza interruzioni di riga.

Le stringhe di esempio possono essere visualizzate su più righe per motivi di leggibilità, ma l'operazione richiede che i parametri vengano inviati come stringhe a riga singola.

```
aws verifiedpermissions put-schema --definition file: //schema.txt -- policy-store-id
PSEXAMPLEabcdefg111111
```

Contenuto di schema.txt.

```
{
  "cedarJson": "{\\"MySampleNamespace\\": {\\"actions\\": {\\"remoteAccess\\": {
    \\"appliesTo\\": {\\"principalTypes\\": [\\"Employee\\"]}},\\"entityTypes\\": {
    \\"Employee\\": {\\"shape\\": {\\"attributes\\": {\\"jobLevel\\": {\\"type\\":
    \\"Long\\"},\\"name\\": {\\"type\\": \\"String\\"}},\\"type\\": \\"Record\\"}}}}}"
}
```

Output:

```
{
  "policyStoreId": "PSEXAMPLEabcdefg111111",
  "namespaces": [
    "MySampleNamespace"
  ],
  "createdDate": "2023-06-14T17:47:13.999885+00:00",
  "lastUpdatedDate": "2023-06-14T17:47:13.999885+00:00"
}
```

Per ulteriori informazioni sullo schema, consulta lo [schema del Policy store](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta [PutSchema AWS CLI](#) Command Reference.

## update-identity-source

Il seguente esempio di codice mostra come utilizzare update-identity-source.

## AWS CLI

Per aggiornare una fonte di identità

L'update-identity-source seguente modifica l'origine dell'identità specificata fornendo una nuova configurazione del pool di utenti di Cognito e cambiando il tipo di entità restituito dall'origine dell'identità.

```
aws verifiedpermissions update-identity-source
  --identity-source-id ISEXAMPLEEabcdefg111111 \
  --update-configuration file://config.txt \
  --principal-entity-type "Employee" \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Contenuto di config.txt.

```
{
  "cognitoUserPoolConfiguration": {
    "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/
us-west-2_1a2b3c4d5",
    "clientIds": ["a1b2c3d4e5f6g7h8i9j0kalbmc"]
  }
}
```

Output:

```
{
  "createdDate": "2023-05-19T20:30:28.214829+00:00",
  "identitySourceId": "ISEXAMPLEEabcdefg111111",
  "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
  "policyStoreId": "PSEXAMPLEEabcdefg111111"
}
```

Per ulteriori informazioni sulle fonti di identità, consulta [Using Amazon Verified Permissions with identity provider](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdateIdentitySource](#) Reference.

## update-policy-store

Il seguente esempio di codice mostra come utilizzare update-policy-store.

## AWS CLI

Per aggiornare un archivio di politiche

L'update-policy-storeesempio seguente modifica un policy store cambiandone l'impostazione di convalida.

```
aws verifiedpermissions update-policy-store \  
  --validation-settings "mode=STRICT" \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Output:

```
{  
  "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/  
PSEXAMPLEEabcdefg111111",  
  "createdDate": "2023-05-16T17:41:29.103459+00:00",  
  "lastUpdatedDate": "2023-05-16T17:41:29.103459+00:00",  
  "policyStoreId": "PSEXAMPLEEabcdefg111111"  
}
```

Per ulteriori informazioni sugli archivi delle politiche, consulta gli archivi [delle politiche di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdatePolicyStoreReference](#).

## update-policy-template

Il seguente esempio di codice mostra come utilizzareupdate-policy-template.

### AWS CLI

Esempio 1: aggiornare un modello di policy

L'update-policy-templateesempio seguente modifica la politica specificata collegata al modello per sostituire la relativa dichiarazione di politica.

```
aws verifiedpermissions update-policy-template \  
  --policy-template-id PTEXAMPLEEabcdefg111111 \  
  --statement file://template1.txt \  
  --policy-store-id PSEXAMPLEEabcdefg111111
```



## Contenuto del file: template1.txt

```
permit(  
  principal in ?principal,  
  action == Action::"view",  
  resource == Photo::"VacationPhoto94.jpg"  
);
```

## Output:

```
{  
  "createdDate": "2023-06-12T20:47:42.804511+00:00",  
  "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",  
  "policyStoreId": "PSEXAMPLEabcdefg111111",  
  "policyTemplateId": "PTEXAMPLEabcdefg111111"  
}
```

Per ulteriori informazioni sui modelli di policy, consulta i modelli di [policy di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdatePolicyTemplate](#)Reference.

## update-policy

Il seguente esempio di codice mostra come utilizzare `update-policy`.

### AWS CLI

Esempio 1: creare una politica statica

L'`create-policy`esempio seguente crea una politica statica con un ambito di policy che specifica sia un principale che una risorsa.

```
aws verifiedpermissions create-policy \  
  --definition file://definition.txt \  
  --policy-store-id PSEXAMPLEabcdefg111111
```

Il `statement` parametro accetta una rappresentazione in formato stringa di un JSON oggetto. Contiene virgolette incorporate («) all'interno della coppia di virgolette più esterna. A tale scopo è necessario JSON convertire il in stringa facendo precedere tutte le virgolette incorporate da una barra rovesciata (") e combinando tutte le righe in un'unica riga di testo senza interruzioni di riga.

Le stringhe di esempio possono essere visualizzate su più righe per motivi di leggibilità, ma l'operazione richiede che i parametri vengano inviati come stringhe a riga singola.

Contenuto del file: `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\" );"
  }
}
```

Output:

```
{
  "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"
  }
}
```

Esempio 2: creare una politica statica che garantisca l'accesso a una risorsa a tutti

L'`create-policy` seguente crea una politica statica con un ambito di policy che specifica solo una risorsa.

```
aws verifiedpermissions create-policy \
  --definition file://definition2.txt \
  --policy-store-id PSEXAMPLEabcdefg111111
```

Contenuto del file `definition2.txt`:

```
{
  "static": {
    "description": "Grant everyone access to the publicFolder Album",
    "statement": "permit(principal, action, resource in Album:\""publicFolder
  \");"
  }
}
```

Output:

```
{
  "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"
  }
}
```

Esempio 3: creare una politica collegata al modello associata al modello specificato

L'`create-policy` seguente crea una politica collegata al modello utilizzando il modello di policy specificato e associa il principale specificato da utilizzare alla nuova politica collegata al modello.

```
aws verifiedpermissions create-policy \
  --definition file://definition2.txt \
  --policy-store-id PSEXAMPLEEabcdefg111111
```

Contenuto di `definition3.txt`:

```
{
  "templateLinked": {
    "policyTemplateId": "PTEXAMPLEEabcdefg111111",
    "principal": {
      "entityType": "User",
      "entityId": "alice"
    }
  }
}
```

```
}  
}
```

Output:

```
{  
  "createdDate": "2023-06-12T20:49:51.490211+00:00",  
  "lastUpdatedDate": "2023-06-12T20:49:51.490211+00:00",  
  "policyId": "TPEXAMPLEabcdefg111111",  
  "policyStoreId": "PSEXAMPLEabcdefg111111",  
  "policyType": "TEMPLATE_LINKED",  
  "principal": {  
    "entityId": "alice",  
    "entityType": "User"  
  },  
  "resource": {  
    "entityId": "VacationPhoto94.jpg",  
    "entityType": "Photo"  
  }  
}
```

Per ulteriori informazioni sulle politiche, consulta le [politiche di Amazon Verified Permissions](#) nella Amazon Verified Permissions User Guide.

- Per API i dettagli, consulta AWS CLI Command [UpdatePolicy](#)Reference.

## VPCEsempi di reticoli utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with VPC Lattice.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

## Azioni

### **create-listener**

Il seguente esempio di codice mostra come utilizzare `create-listener`.

#### AWS CLI

Per creare un ascoltatore

L'`create-listener` esempio seguente crea un HTTPS listener con una regola predefinita che inoltra il traffico al gruppo target VPC Lattice specificato.

```
aws vpc-lattice create-listener \  
  --name my-service-listener \  
  --protocol HTTPS \  
  --port 443 \  
  --service-identifier svc-0285b53b2eEXAMPLE \  
  --default-action file://listener-config.json
```

Contenuto di `listener-config.json`.

```
{  
  "forward": {  
    "targetGroups": [  
      {  
        "targetGroupIdentifier": "tg-0eaa4b9ab4EXAMPLE"  
      }  
    ]  
  }  
}
```

Output:

```
{  
  "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"
}

```

Per ulteriori informazioni, consulta [Listeners](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateListener](#) Reference.

## create-service-network-service-association

Il seguente esempio di codice mostra come utilizzare `create-service-network-service-association`.

### AWS CLI

Per creare un'associazione di servizi

L'`create-service-network-service-association` seguente associa il servizio specificato alla rete di servizi specificata.

```

aws vpc-lattice create-service-network-service-association \
  --service-identifier svc-0285b53b2eEXAMPLE \
  --service-network-identifier sn-080ec7dc93EXAMPLE

```

Output:

```

{
  "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"
}

```

Per ulteriori informazioni, consulta [Gestire le associazioni di servizi](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [CreateServiceNetworkServiceAssociation AWS CLI Command Reference](#).

## create-service-network-vpc-association

Il seguente esempio di codice mostra come utilizzare `create-service-network-vpc-association`.

### AWS CLI

Per creare un'VPC associazione

L'`create-service-network-vpc-association` esempio seguente associa il vpc specificato alla rete di servizi specificata. Il gruppo di sicurezza specificato controlla quali risorse in VPC possono accedere alla rete di servizi e ai relativi servizi.

```

aws vpc-lattice create-service-network-vpc-association \
  --vpc-identifier vpc-0a1b2c3d4eEXAMPLE \
  --service-network-identifier sn-080ec7dc93EXAMPLE \
  --security-group-ids sg-0aee16bc6cEXAMPLE

```

Output:

```

{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/
snva-0821fc8631EXAMPLE",
  "createdBy": "123456789012",
  "id": "snva-0821fc8631EXAMPLE",
  "securityGroupIds": [
    "sg-0aee16bc6cEXAMPLE"
  ]
}

```

```
  ],  
  "status": "CREATE_IN_PROGRESS"  
}
```

Per ulteriori informazioni, consulta [Gestire VPC le associazioni](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [CreateServiceNetworkVpcAssociation AWS CLI Command Reference](#).

## **create-service-network**

Il seguente esempio di codice mostra come utilizzare `create-service-network`.

### AWS CLI

Per creare una rete di servizi

L'`create-service-network` esempio seguente crea una rete di servizi con il nome specificato.

```
aws vpc-lattice create-service-network \  
  --name my-service-network
```

Output:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/  
sn-080ec7dc93EXAMPLE",  
  "authType": "NONE",  
  "id": "sn-080ec7dc93EXAMPLE",  
  "name": "my-service-network"  
}
```

Per ulteriori informazioni, consulta [Service networks](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [CreateServiceNetwork AWS CLI Command Reference](#).

## **create-service**

Il seguente esempio di codice mostra come utilizzare `create-service`.



## AWS CLI

Per creare un servizio

L'`create-service` seguente crea un servizio con il nome specificato.

```
aws vpc-lattice create-service \  
  --name my-lattice-service
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Services in VPC Lattice](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta AWS CLI Command [CreateService](#)Reference.

## `create-target-group`

Il seguente esempio di codice mostra come utilizzare `create-target-group`.

### AWS CLI

Esempio 1: Per creare un gruppo target di tipo INSTANCE

L'`create-target-group` seguente crea un gruppo target con il nome, il tipo e la configurazione specificati.

```
aws vpc-lattice create-target-group \  
  --name my-lattice-target-group
```

```
--name my-lattice-target-group-instance \  
--type INSTANCE \  
--config file://tg-config.json
```

Contenuto di tg-config.json.

```
{  
  "port": 443,  
  "protocol": "HTTPS",  
  "protocolVersion": "HTTP1",  
  "vpcIdentifier": "vpc-f1663d9868EXAMPLE"  
}
```

Output:

```
{  
  "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"  
}
```

## Esempio 2: creare un gruppo target di tipo IP

L'`create-target-group` seguente crea un gruppo target con il nome, il tipo e la configurazione specificati.

```
aws vpc-lattice create-target-group \  
  --name my-lattice-target-group-ip \  
  --type IP \  
  --config file://tg-config.json
```

Contenuto di `tg-config.json`.

```
{  
  "ipAddressType": "IPV4",  
  "port": 443,  
  "protocol": "HTTPS",  
  "protocolVersion": "HTTP1",  
  "vpcIdentifier": "vpc-f1663d9868EXAMPLE"  
}
```

Output:

```
{  
  "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"
}
```

### Esempio 3: Per creare un gruppo target di tipo LAMBDA

L'`create-target-group` seguente crea un gruppo target con il nome, il tipo e la configurazione specificati.

```
aws vpc-lattice create-target-group \
  --name my-lattice-target-group-lambda \
  --type LAMBDA
```

Output:

```
{
  "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"
}
```

### Esempio 4: Per creare un gruppo target di tipo ALB

L'`create-target-group` seguente crea un gruppo target con il nome, il tipo e la configurazione specificati.

```
aws vpc-lattice create-target-group \
  --name my-lattice-target-group-alb \
  --type ALB \
  --config file://tg-config.json
```

Contenuto di `tg-config.json`.

```
{
  "port": 443,
  "protocol": "HTTPS",
  "protocolVersion": "HTTP1",
  "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Target groups](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [CreateTargetGroup AWS CLI Command Reference](#).

## delete-auth-policy

Il seguente esempio di codice mostra come utilizzare `delete-auth-policy`.

### AWS CLI

Per eliminare una politica di autenticazione

L'`delete-auth-policy`esempio seguente elimina la politica di autenticazione per il servizio specificato.

```
aws vpc-lattice delete-auth-policy \
  --resource-identifier svc-0285b53b2eEXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta le [politiche di autenticazione](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteAuthPolicy](#) Reference.

## **delete-listener**

Il seguente esempio di codice mostra come utilizzare `delete-listener`.

### AWS CLI

Per eliminare un ascoltatore

L'`delete-listener` esempio seguente elimina il listener specificato.

```
aws vpc-lattice delete-listener \  
  --listener-identifier listener-07cc7fb0abEXAMPLE \  
  --service-identifier svc-0285b53b2eEXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Listeners](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteListener](#) Reference.

## **delete-service-network-service-association**

Il seguente esempio di codice mostra come utilizzare `delete-service-network-service-association`.

### AWS CLI

Per eliminare un'associazione di servizi

L'`delete-service-network-service-association` esempio seguente dissocia l'associazione di servizi specificata.

```
aws vpc-lattice delete-service-network-service-association \  
  --service-network-service-association-identifier snsa-031fabb4d8EXAMPLE
```

**Output:**

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkserviceassociation/snsa-031fabb4d8EXAMPLE",
  "id": "snsa-031fabb4d8EXAMPLE",
  "status": "DELETE_IN_PROGRESS"
}
```

Per ulteriori informazioni, consulta [Gestire le associazioni di servizi](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [DeleteServiceNetworkServiceAssociation AWS CLI Command Reference](#).

**delete-service-network-vpc-association**

Il seguente esempio di codice mostra come utilizzare `delete-service-network-vpc-association`.

**AWS CLI**

Per eliminare un'VPC associazione

L'`delete-service-network-vpc-association` esempio seguente dissocia l'associazione specificata VPC.

```
aws vpc-lattice delete-service-network-vpc-association \
  --service-network-vpc-association-identifier snva-0821fc8631EXAMPLE
```

**Output:**

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/snva-0821fc8631EXAMPLE",
  "id": "snva-0821fc8631EXAMPLE",
  "status": "DELETE_IN_PROGRESS"
}
```

Per ulteriori informazioni, consulta [Gestire VPC le associazioni](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [DeleteServiceNetworkVpcAssociation AWS CLI Command Reference](#).

## delete-service-network

Il seguente esempio di codice mostra come utilizzare `delete-service-network`.

### AWS CLI

Per eliminare una rete di servizi

L'`delete-service-network` esempio seguente elimina la rete di servizi specificata.

```
aws vpc-lattice delete-service-network \  
  --service-network-identifier sn-080ec7dc93EXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Service networks](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [DeleteServiceNetwork AWS CLI Command Reference](#).

## delete-service

Il seguente esempio di codice mostra come utilizzare `delete-service`.

### AWS CLI

Per eliminare un servizio

L'`delete-service` esempio seguente elimina il servizio specificato.

```
aws vpc-lattice delete-service \  
  --service-identifier svc-0285b53b2eEXAMPLE
```

Output:

```
{  
  "arn": "arn:aws:vpc-lattice:us-west-2:123456789012:service/  
svc-0285b53b2eEXAMPLE",  
  "id": "svc-0285b53b2eEXAMPLE",  
  "name": "my-lattice-service",
```



```
"status": "DELETE_IN_PROGRESS"
}
```

Per ulteriori informazioni, consulta [Services in VPC Lattice](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta AWS CLI Command [DeleteServiceReference](#).

## **delete-target-group**

Il seguente esempio di codice mostra come utilizzare `delete-target-group`.

### AWS CLI

Per eliminare un gruppo target

L'`delete-target-group`esempio seguente elimina il gruppo target specificato.

```
aws vpc-lattice delete-target-group \
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Output:

```
{
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
  "id": "tg-0eaa4b9ab4EXAMPLE",
  "status": "DELETE_IN_PROGRESS"
}
```

Per ulteriori informazioni, consulta [Target groups](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [DeleteTargetGroup AWS CLI](#)Command Reference.

## **deregister-targets**

Il seguente esempio di codice mostra come utilizzare `deregister-targets`.

### AWS CLI

Annullare la registrazione di un bersaglio

L'`deregister-targets`esempio seguente annulla la registrazione della destinazione specificata dal gruppo di destinazione specificato.

```
aws vpc-lattice deregister-targets \  
  --targets i-07dd579bc5EXAMPLE \  
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Output:

```
{  
  "successful": [  
    {  
      "id": "i-07dd579bc5EXAMPLE",  
      "port": 443  
    }  
  ],  
  "unsuccessful": []  
}
```

Per ulteriori informazioni, consulta [Registrazione degli obiettivi](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [DeregisterTargets AWS CLI Command Reference](#).

## get-auth-policy

Il seguente esempio di codice mostra come utilizzare `get-auth-policy`.

### AWS CLI

Per ottenere informazioni su una politica di autenticazione

L'esempio seguente ottiene informazioni sulla politica di autenticazione per il servizio specificato.

```
aws vpc-lattice get-auth-policy \  
  --resource-identifier svc-0285b53b2eEXAMPLE
```

Output:

```
{  
  "createdAt": "2023-06-07T03:51:20.266Z",  
  "lastUpdatedAt": "2023-06-07T04:39:27.082Z",  
  "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"
}
```

Per ulteriori informazioni, consulta le [politiche di autenticazione](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetAuthPolicyReference](#).

## get-listener

Il seguente esempio di codice mostra come utilizzare `get-listener`.

### AWS CLI

Per ottenere informazioni su un service listener

L'`get-listener` esempio seguente ottiene informazioni sul listener specificato per il servizio specificato.

```
aws vpc-lattice get-listener \
  --listener-identifier listener-0ccf55918cEXAMPLE \
  --service-identifier svc-0285b53b2eEXAMPLE
```

Output:

```
{
  "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": [
        {
          "targetGroupIdentifier": "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"
}
```

Per ulteriori informazioni, consulta [Define routing](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta AWS CLI Command [GetListenerReference](#).

## get-service-network-service-association

Il seguente esempio di codice mostra come utilizzare `get-service-network-service-association`.

### AWS CLI

Per ottenere informazioni su un'associazione di servizi

L'`get-service-network-service-association` esempio seguente ottiene informazioni sull'associazione di servizi specificata.

```
aws vpc-lattice get-service-network-service-association \
  --service-network-service-association-identifier snsa-031fabb4d8EXAMPLE
```

Output:

```
{
  "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"
}

```

Per ulteriori informazioni, consulta [Gestire le associazioni di servizi](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [GetServiceNetworkServiceAssociation AWS CLI Command Reference](#).

## get-service-network-vpc-association

Il seguente esempio di codice mostra come utilizzare `get-service-network-vpc-association`.

### AWS CLI

Per ottenere informazioni su un'VPC associazione

L'`get-service-network-vpc-association` esempio seguente ottiene informazioni sull'VPC associazione specificata.

```

aws vpc-lattice get-service-network-vpc-association \
--service-network-vpc-association-identifier snva-0821fc8631EXAMPLE

```

Output:

```

{
  "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"
}
```

Per ulteriori informazioni, consulta [Gestire VPC le associazioni](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [GetServiceNetworkVpcAssociation AWS CLI](#) Command Reference.

## get-service-network

Il seguente esempio di codice mostra come utilizzare `get-service-network`.

### AWS CLI

Per ottenere informazioni su una rete di servizi

L'`get-service-network` esempio seguente ottiene informazioni sulla rete di servizi specificata.

```
aws vpc-lattice get-service-network \
--service-network-identifier sn-080ec7dc93EXAMPLE
```

Output:

```
{
  "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
}
```

Per ulteriori informazioni, consulta [Service networks](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [GetServiceNetwork AWS CLI](#) Command Reference.

## get-service

Il seguente esempio di codice mostra come utilizzare `get-service`.

### AWS CLI

Per ottenere informazioni su un servizio

L'`get-service` esempio seguente ottiene informazioni sul servizio specificato.

```
aws vpc-lattice get-service \  
  --service-identifier svc-0285b53b2eEXAMPLE
```

Output:

```
{  
  "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": "Z09127221KTH2CFUOHIZH"  
  },  
  "id": "svc-0285b53b2eEXAMPLE",  
  "lastUpdatedAt": "2023-05-05T21:35:29.339Z",  
  "name": "my-lattice-service",  
  "status": "ACTIVE"  
}
```

Per ulteriori informazioni, consulta la sezione [Servizi](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [GetService AWS CLI Command Reference](#).

## get-target-group

Il seguente esempio di codice mostra come utilizzare `get-target-group`.

### AWS CLI

Per ottenere informazioni su un gruppo target

L'get-target-groupesempio seguente ottiene informazioni sul gruppo target specificato, che ha un tipo di destinazione diINSTANCE.

```
aws vpc-lattice get-target-group \  
--target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, consulta [Target groups](#) nella Amazon VPC Lattice User Guide.



- Per API i dettagli, consulta [GetTargetGroup AWS CLI Command Reference](#).

## list-listeners

Il seguente esempio di codice mostra come utilizzare `list-listeners`.

### AWS CLI

Per elencare gli ascoltatori del servizio

L'`list-listeners` esempio seguente elenca i listener per il servizio specificato.

```
aws vpc-lattice list-listeners \  
  --service-identifier svc-0285b53b2eEXAMPLE
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Define routing](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta AWS CLI Command [ListListeners](#) Reference.

## list-service-network-service-associations

Il seguente esempio di codice mostra come utilizzare `list-service-network-service-associations`.

## AWS CLI

Per elencare le associazioni di servizi

L'`list-service-network-service-associations` esempio seguente elenca le associazioni di servizi per la rete di servizi specificata. L'`--query` opzione indirizza l'output alle associazioni IDs di servizi.

```
aws vpc-lattice list-service-network-service-associations \  
  --service-network-identifier sn-080ec7dc93EXAMPLE \  
  --query items[*].id
```

Output:

```
[  
  "snsa-031fabb4d8EXAMPLE",  
  "snsa-0e16955a8cEXAMPLE"  
]
```

Per ulteriori informazioni, consulta [Gestire le associazioni di servizi](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [ListServiceNetworkServiceAssociations AWS CLI Command Reference](#).

## `list-service-network-vpc-associations`

Il seguente esempio di codice mostra come utilizzare `list-service-network-vpc-associations`.

## AWS CLI

Per elencare VPC le associazioni

L'`list-service-network-vpc-associations` esempio seguente elenca le VPC associazioni per la rete di servizi specificata. L'`--query` opzione indirizza l'output IDs alle VPC associazioni.

```
aws vpc-lattice list-service-network-vpc-associations \  
  --service-network-identifier sn-080ec7dc93EXAMPLE \  
  --query items[*].id
```

Output:

```
[
  "snva-0821fc8631EXAMPLE",
  "snva-0c5dcb60d6EXAMPLE"
]
```

Per ulteriori informazioni, consulta [Gestire VPC le associazioni](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [ListServiceNetworkVpcAssociations AWS CLI](#) Command Reference.

## list-service-networks

Il seguente esempio di codice mostra come utilizzare `list-service-networks`.

AWS CLI

Per elencare le reti di servizio

L'`list-service-networks` esempio seguente elenca le reti di servizio possedute o condivise con l'account chiamante. L'`--query` opzione assegna i risultati agli Amazon Resource Names (ARN) delle reti di servizi.

```
aws vpc-lattice list-service-networks \
  --query items[*].arn
```

Output:

```
[
  "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
  "arn:aws:vpc-lattice:us-east-2:111122223333:servicenetwork/sn-0ec4d436cfEXAMPLE"
]
```

Per ulteriori informazioni, consulta [Service networks](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [ListServiceNetworks AWS CLI](#) Command Reference.

## list-services

Il seguente esempio di codice mostra come utilizzare `list-services`.

## AWS CLI

Per elencare i tuoi servizi

L'`list-services` esempio seguente elenca i servizi posseduti o condivisi con l'account chiamante. L'`--query` opzione assegna i risultati agli Amazon Resource Names (ARN) dei servizi.

```
aws vpc-lattice list-services \  
  --query items[*].arn
```

Output:

```
[  
  "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE",  
  "arn:aws:vpc-lattice:us-east-2:111122223333:service/svc-0b8ac96550EXAMPLE"  
]
```

Per ulteriori informazioni, consulta la sezione [Servizi](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [ListServices AWS CLI Command Reference](#).

## list-target-groups

Il seguente esempio di codice mostra come utilizzare `list-target-groups`.

### AWS CLI

Per elencare i tuoi gruppi target

L'`list-target-groups` esempio seguente elenca i gruppi target con un tipo di oggetto di LAMBDA.

```
aws vpc-lattice list-target-groups \  
  --target-group-type LAMBDA
```

Output:

```
{  
  "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"
  }
]
}

```

Per ulteriori informazioni, consulta [Target groups](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [ListTargetGroups AWS CLI Command Reference](#).

## list-targets

Il seguente esempio di codice mostra come utilizzare `list-targets`.

### AWS CLI

Per elencare gli obiettivi di un gruppo target

L'`list-targets` esempio seguente elenca le destinazioni per il gruppo target specificato.

```

aws vpc-lattice list-targets \
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE

```

Output:

```

{
  "items": [
    {
      "id": "i-07dd579bc5EXAMPLE",
      "port": 443,
      "status": "HEALTHY"
    },
    {

```

```

        "id": "i-047b3c9078EXAMPLE",
        "port": 443,
        "reasonCode": "HealthCheckFailed",
        "status": "UNHEALTHY"
    }
]
}

```

Per ulteriori informazioni, consulta [Target groups](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [ListTargets AWS CLI](#) Command Reference.

## put-auth-policy

Il seguente esempio di codice mostra come utilizzare `put-auth-policy`.

### AWS CLI

Per creare una politica di autenticazione per un servizio

L'`put-auth-policy` esempio seguente concede l'accesso alle richieste di qualsiasi principale autenticato che utilizza il ruolo specificato. IAM La risorsa è quella ARN del servizio a cui è allegata la policy.

```

aws vpc-lattice put-auth-policy \
  --resource-identifier svc-0285b53b2eEXAMPLE \
  --policy file://auth-policy.json

```

Contenuto di `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"
    }
  ]
}

```

```
]
}
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta le [politiche di autenticazione](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta AWS CLI Command [PutAuthPolicy](#) Reference.

## register-targets

Il seguente esempio di codice mostra come utilizzare `register-targets`.

### AWS CLI

Per registrare un bersaglio

L'esempio seguente registra le destinazioni specificate con il gruppo di destinazione specificato.

```
aws vpc-lattice register-targets \
  --targets id=i-047b3c9078EXAMPLE id=i-07dd579bc5EXAMPLE \
  --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

Output:

```
{
  "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
      }
    ]
  ]
}
```

Per ulteriori informazioni, consulta [Registrazione degli obiettivi](#) nella Amazon VPC Lattice User Guide.

- Per API i dettagli, consulta [RegisterTargets AWS CLI Command Reference](#).

## AWS WAF Classic esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS WAF Classic.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **put-logging-configuration**

Il seguente esempio di codice mostra come utilizzare `put-logging-configuration`.

AWS CLI

Per creare una configurazione di registrazione per il Web ACL ARN con il flusso Kinesis Firehose specificato ARN



L'put-logging-configurationesempio seguente mostra la configurazione di registrazione per with. WAF 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=[]
```

Output:

```
{
  "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"
    ]
  }
}
```

- Per API i dettagli, vedere [PutLoggingConfiguration](#) in AWS CLI Command Reference.

## update-byte-match-set

Il seguente esempio di codice mostra come utilizzare update-byte-match-set.

AWS CLI

Per aggiornare un set di corrispondenze di byte

Il update-byte-match-set comando seguente elimina un ByteMatchTuple oggetto (filtro) in un: 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
```

Per ulteriori informazioni, consulta Working with String Match Conditions nella guida per AWS WAFsviluppatore.

- Per API i dettagli, consulta [UpdateByteMatchSet AWS CLI Command Reference](#).

## update-ip-set

Il seguente esempio di codice mostra come utilizzare `update-ip-set`.

### AWS CLI

Per aggiornare un set IP

Il `update-ip-set` comando seguente aggiorna un IPSet con un IPv4 indirizzo ed elimina un IPv6 indirizzo:

```
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
```

In alternativa è possibile utilizzare un JSON file per specificare l'input. Per esempio:

```
aws waf update-ip-set --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-
token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates file://change.json
```

Dove il contenuto del JSON file è:

```
[
{
  "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"
  }
}
]
```

Per ulteriori informazioni, consulta [Working with IP Match Conditions](#) nella guida per AWS WAFsvilupatori.

- Per API i dettagli, consulta [UpdateIpSet AWS CLI Command Reference](#).

## update-rule

Il seguente esempio di codice mostra come utilizzare `update-rule`.

### AWS CLI

Per aggiornare una regola

Il `update-rule` comando seguente elimina un oggetto Predicate in una regola:

```
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"}
```

Per ulteriori informazioni, consulta [Working with Rules](#) nella guida per AWS WAFsvilupatori.

- Per API i dettagli, consulta [UpdateRule AWS CLI Command Reference](#).

## update-size-constraint-set

Il seguente esempio di codice mostra come utilizzare `update-size-constraint-set`.

### AWS CLI

Per aggiornare un set di vincoli di dimensione

Il `update-size-constraint-set` comando seguente elimina un SizeConstraint oggetto (filtri) in un set di vincoli di dimensione:

```
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"}
```

Per ulteriori informazioni, consulta [Lavorare con le condizioni relative ai vincoli di dimensione](#) nella guida per sviluppatori. AWS WAF

- Per API i dettagli, consulta AWS CLI Command [UpdateSizeConstraintSetReference](#).

## update-sql-injection-match-set

Il seguente esempio di codice mostra come utilizzare `update-sql-injection-match-set`.

### AWS CLI

Per aggiornare un SQL Injection Match Set

Il `update-sql-injection-match-set` comando seguente elimina un `SqliInjectionMatchTuple` oggetto (filtri) in un set di match di SQL iniezione:

```
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"},TextTransformati
```

Per ulteriori informazioni, consulta [Working with SQL Injection Match Conditions](#) nella guida per AWS WAFsviluppatore.

- Per API i dettagli, vedere [UpdateSqlInjectionMatchSet](#) in AWS CLI Command Reference.

## update-web-acl

Il seguente esempio di codice mostra come utilizzare `update-web-acl`.

### AWS CLI

Per aggiornare un Web ACL

Il `update-web-acl` comando seguente elimina un `ActivatedRule` oggetto in un WebACL.

```
aws waf update-web-acl -- web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-token  
12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates Azione=» «, =' {Priority=1, =» -1-Esempio»,  
Azione= {Tipo=» «}, Tipo=» «}» DELETE ActivatedRule RuleId WAFRule ALLOW REGULAR
```

Output:

```
{  
  "ChangeToken": "12cs345-67cd-890b-1cd2-c3a4567d89f1"
```

```
}
```

Per ulteriori informazioni, consulta [Working with Web ACLs](#) nella AWS WAF, AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [UpdateWebAcl](#) in AWS CLI Command Reference.

## update-xss-match-set

Il seguente esempio di codice mostra come utilizzare `update-xss-match-set`.

### AWS CLI

Per aggiornare un XSSMatchSet

Il `update-xss-match-set` comando seguente elimina un XssMatchTuple oggetto (filtri) in un 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
```

Per ulteriori informazioni, consulta [Working with Cross-site Scripting Match Conditions](#) nella AWS WAF guida per sviluppatori.

- Per API i dettagli, consulta AWS CLI Command [UpdateXssMatchSet](#) Reference.

## AWS WAF Classic regionale esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS WAF Classic regionale.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

## Azioni

### **associate-web-acl**

Il seguente esempio di codice mostra come utilizzare `associate-web-acl`.

#### AWS CLI

Associare un Web ACL a una risorsa

Il `associate-web-acl` comando seguente associa un WebACL, specificato da `web-acl-id`, a una risorsa specificata da `resource-arn`. La risorsa ARN può fare riferimento a un Application Load Balancer o a un Gateway: API

```
aws waf-regional associate-web-acl \  
  --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \  
  --resource-arn 12cs345-67cd-890b-1cd2-c3a4567d89f1
```

Per ulteriori informazioni, consulta [Working with Web ACLs](#) nella AWS WAF Developer Guide.

- Per API i dettagli, vedere [AssociateWebAcl](#) in AWS CLI Command Reference.

### **put-logging-configuration**

Il seguente esempio di codice mostra come utilizzare `put-logging-configuration`.

#### AWS CLI

Per creare una configurazione di registrazione per il Web ACL ARN con il flusso Kinesis Firehose specificato ARN

L'esempio seguente mostra la configurazione di registrazione per WAF with ALB/in Region. API Gateway 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
```

## Output:

```
{
  "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"
    ]
  }
}
```

- Per API i dettagli, vedere [PutLoggingConfiguration](#) in AWS CLI Command Reference.

## update-byte-match-set

Il seguente esempio di codice mostra come utilizzare `update-byte-match-set`.

### AWS CLI

Per aggiornare un set di corrispondenze di byte

Il `update-byte-match-set` comando seguente elimina un `ByteMatchTuple` oggetto (filtro) in un `ByteMatchSet`. Poiché il `updates` valore contiene virgolette doppie incorporate, è necessario racchiuderlo tra virgolette singole.

```
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="
```

Per ulteriori informazioni, consulta [Working with String Match Conditions](#) nella Guida per gli AWS WAF sviluppatori.

- Per API i dettagli, consulta [UpdateByteMatchSet AWS CLI](#) Command Reference.

## update-ip-set

Il seguente esempio di codice mostra come utilizzare `update-ip-set`.

## AWS CLI

Per aggiornare un set IP

Il `update-ip-set` comando seguente aggiorna un IPSet con un IPv4 indirizzo ed elimina un IPv6 indirizzo. Ottieni il valore per `change-token` eseguendo il `get-change-token` comando. Poiché il valore per gli aggiornamenti include virgolette doppie incorporate, è necessario racchiudere il valore tra virgolette singole.

```
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
```

In alternativa, è possibile utilizzare un JSON file per specificare l'input. Per esempio:

```
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
```

Contenuto del `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"  
    }  
  }  
]
```



```
]
```

Per ulteriori informazioni, consulta [Working with IP Match Conditions](#) nella AWS WAF Developer Guide.

- Per API i dettagli, consulta [UpdateIpSet AWS CLI Command Reference](#).

## update-rule

Il seguente esempio di codice mostra come utilizzare `update-rule`.

### AWS CLI

Per aggiornare una regola

Il `update-rule` comando seguente elimina un Predicate oggetto in una regola. Poiché il `updates` valore contiene virgolette doppie incorporate, è necessario racchiudere l'intero valore tra virgolette singole.

```
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"}'
```

Per ulteriori informazioni, consulta [Working with Rules](#) nella AWS WAF Developer Guide.

- Per API i dettagli, vedere [UpdateRule](#) in AWS CLI Command Reference.

## update-size-constraint-set

Il seguente esempio di codice mostra come utilizzare `update-size-constraint-set`.

### AWS CLI

Per aggiornare un set di vincoli di dimensione

Il `update-size-constraint-set` comando seguente elimina un oggetto `SizeConstraint` (filtri) in un set di vincoli di dimensione. Poiché il `updates` valore contiene virgolette doppie incorporate, è necessario racchiudere l'intero valore tra virgolette singole.

```
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"}
```

Per ulteriori informazioni, consulta [Lavorare con le condizioni relative ai vincoli di dimensione nella Guida](#) per gli AWS WAF sviluppatori.

- Per API i dettagli, consulta AWS CLI Command [UpdateSizeConstraintSetReference](#).

## update-sql-injection-match-set

Il seguente esempio di codice mostra come utilizzare `update-sql-injection-match-set`.

### AWS CLI

Per aggiornare un SQL Injection Match Set

Il `update-sql-injection-match-set` comando seguente elimina un `SqlInjectionMatchTuple` oggetto (filtri) in un set di SQL Injection Match. Poiché il `updates` valore contiene virgolette doppie incorporate, è necessario racchiudere l'intero valore tra virgolette singole. :

```
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 'Azione=> <, = {Tipo => _ <<}, => _ <<}'
'DELETESqlInjectionMatchTupleFieldToMatchQUERYSTRINGTextTransformationURLDECODE'
```

Per ulteriori informazioni, consulta [Working with SQL Injection Match Conditions](#) nella Guida per gli AWS WAF sviluppatori.

- Per API i dettagli, consulta [UpdateSqlInjectionMatchSet AWS CLI Command Reference](#).

## update-web-acl

Il seguente esempio di codice mostra come utilizzare `update-web-acl`.

### AWS CLI

Per aggiornare un sito web ACL

Il `update-web-acl` comando seguente elimina un `ActivatedRule` oggetto in un `WebACL`. Poiché il `updates` valore contiene virgolette doppie incorporate, è necessario racchiudere l'intero valore tra virgolette singole.

```
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"}'
```

Per ulteriori informazioni, consulta [Working with Web ACLs](#) nella AWS WAF Developer Guide.

- Per API i dettagli, vedere [UpdateWebAcl](#) in AWS CLI Command Reference.

## update-xss-match-set

Il seguente esempio di codice mostra come utilizzare `update-xss-match-set`.

### AWS CLI

Per aggiornare un `XSSMatchSet`

Il `update-xss-match-set` comando seguente elimina un `XssMatchTuple` oggetto (filtri) in un `XssMatchSet`. Poiché il `updates` valore contiene virgolette doppie incorporate, è necessario racchiudere l'intero valore tra virgolette singole.

```
aws waf-regional 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_
```

Per ulteriori informazioni, consulta [Working with Cross-Site Scripting Match Conditions](#) nella Guida per gli AWS WAF sviluppatori.

- Per API i dettagli, consulta AWS CLI Command [UpdateXssMatchSet](#) Reference.

## AWS WAFV2 esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with AWS WAFV2.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

## Azioni

### **associate-web-acl**

Il seguente esempio di codice mostra come utilizzare `associate-web-acl`.

AWS CLI

Associare un Web ACL a una AWS risorsa regionale

L'esempio seguente associa il Web ACL specificato a un 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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Associare o dissociare un Web ACL con una AWS risorsa](#) nella AWS Firewall AWS WAF Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [AssociateWebAcl](#) in AWS CLI Command Reference.

### **check-capacity**

Il seguente esempio di codice mostra come utilizzare `check-capacity`.

## AWS CLI

Per ottenere la capacità utilizzata da un insieme di regole

Di seguito `check-capacity` vengono recuperati i requisiti di capacità per un set di regole che contiene un'istruzione basata sulla frequenza e un'istruzione di regola che contiene AND regole annidate.

```
aws wafv2 check-capacity \  
  --scope REGIONAL \  
  --rules file://waf-rule-list.json \  
  --region us-west-2
```

Contenuto del 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"
  }
}
]

```

Output:

```
{
```

```
"Capacity":15
}
```

Per ulteriori informazioni, vedere [AWS WAF Web ACL Capacity Units \(WCU\)](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [CheckCapacity](#) in AWS CLI Command Reference.

## create-ip-set

Il seguente esempio di codice mostra come utilizzare `create-ip-set`.

### AWS CLI

Per creare un set IP da utilizzare nel Web ACLs e nei gruppi di regole

Il `create-ip-set` comando seguente crea un set IP con una specifica di intervallo di indirizzi singolo.

```
aws wafv2 create-ip-set \
  --name testip \
  --scope REGIONAL \
  --ip-address-version IPV4 \
  --addresses 198.51.100.0/16
```

Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, vedere [IP Sets and Regex Pattern Sets](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [CreateIpSet](#) in AWS CLI Command Reference.

## create-regex-pattern-set

Il seguente esempio di codice mostra come utilizzare `create-regex-pattern-set`.

### AWS CLI

Per creare un set di pattern regex da utilizzare nel Web ACLs e nei gruppi di regole

Il `create-regex-pattern-set` comando seguente crea un set di pattern regex con due modelli regex specificati.

```
aws wafv2 create-regex-pattern-set \  
  --name regexPatterSet01 \  
  --scope REGIONAL \  
  --description 'Test web-acl' \  
  --regular-expression-list '["RegexString": "/[0-9]*/"], {"RegexString": "/[a-z]*/"}']
```

Output:

```
{  
  "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"  
  }  
}
```

Per ulteriori informazioni, vedere [IP Sets and Regex Pattern Sets](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [CreateRegexPatternSet](#) in AWS CLI Command Reference.

## create-rule-group

Il seguente esempio di codice mostra come utilizzare `create-rule-group`.

### AWS CLI

Per creare un gruppo di regole personalizzato da utilizzare nel Web ACLs



Il `create-rule-group` comando seguente crea un gruppo di regole personalizzato per uso regionale. Le istruzioni delle regole per il gruppo sono fornite in un file in JSON formato.

```
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
```

Contenuto del file: `//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"
}
}
]

```

Output:

```

{
  "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"
  }
}

```

Per ulteriori informazioni, vedere [Managing Your Own Rule Group](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [CreateRuleGroup](#) in AWS CLI Command Reference.

## create-web-acl

Il seguente esempio di codice mostra come utilizzare `create-web-acl`.

## AWS CLI

Per creare un web ACL

Il `create-web-acl` comando seguente crea un Web ACL per uso regionale. Le istruzioni delle regole per il Web ACL sono fornite in un file in JSON formato.

```
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
```

Contenuto del file: `//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"
}
}
```

#### Output:

```
{
  "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"
  }
}
```

Per ulteriori informazioni, vedere [Managing and Using a Web Access Control List \(WebACL\)](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [CreateWebAcl](#) in AWS CLI Command Reference.

## delete-ip-set

Il seguente esempio di codice mostra come utilizzare `delete-ip-set`.

### AWS CLI

Per eliminare un set IP

Quanto segue `delete-ip-set` elimina il set IP specificato. Questa chiamata richiede un ID, che è possibile ottenere dalla chiamata `list-ip-sets`, e un token di blocco, che è possibile ottenere dalle chiamate, `list-ip-sets` e `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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [IP Sets and Regex Pattern Sets](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [DeleteIpSet](#) in AWS CLI Command Reference.

## delete-logging-configuration

Il seguente esempio di codice mostra come utilizzare `delete-logging-configuration`.

### AWS CLI

Per disabilitare la registrazione per un Web ACL

Quanto segue `delete-logging-configuration` rimuove qualsiasi configurazione di registrazione dal Web specificato. ACL

```
aws wafv2 delete-logging-configuration \  
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Logging Web ACL Traffic Information](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [DeleteLoggingConfiguration](#) in AWS CLI Command Reference.

## delete-regex-pattern-set

Il seguente esempio di codice mostra come utilizzare `delete-regex-pattern-set`.

### AWS CLI

Per eliminare un set di pattern regex

Quanto segue `delete-regex-pattern-set` aggiorna le impostazioni per il set di pattern regex specificato. Questa chiamata richiede un ID, che è possibile ottenere dalla chiamata `list-regex-pattern-sets`, e un token di blocco, che è possibile ottenere dalla chiamata `list-regex-pattern-sets` o dalla chiamata `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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [IP Sets and Regex Pattern Sets](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [DeleteRegexPatternSet](#) in AWS CLI Command Reference.

## delete-rule-group

Il seguente esempio di codice mostra come utilizzare `delete-rule-group`.

### AWS CLI

Per eliminare un gruppo di regole personalizzato

Quanto segue `delete-rule-group` elimina il gruppo di regole personalizzate specificato. Questa chiamata richiede un ID, che è possibile ottenere dalla chiamata `list-rule-groups`,

e un token di blocco, che è possibile ottenere dalla chiamata `list-rule-groups` o dalla chiamata `get-rule-group`.

```
aws wafv2 delete-rule-group \  
  --name TestRuleGroup \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --lock-token 7b3bcec2-0000-0000-0000-563bf47249f0
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Managing Your Own Rule Group](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [DeleteRuleGroup](#) in AWS CLI Command Reference.

## delete-web-acl

Il seguente esempio di codice mostra come utilizzare `delete-web-acl`.

### AWS CLI

Per eliminare un Web ACL

Quanto segue `delete-web-acl` elimina il Web specificato ACL dal tuo account. Un Web ACL può essere eliminato solo quando non è associato ad alcuna risorsa. Questa chiamata richiede un ID, che è possibile ottenere dalla chiamata `list-web-acls`, e un token di blocco, che è possibile ottenere dalla chiamata `list-web-acls` o dalla chiamata `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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Managing and Using a Web Access Control List \(WebACL\)](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [DeleteWebAcl](#) in AWS CLI Command Reference.

## describe-managed-rule-group

Il seguente esempio di codice mostra come utilizzare `describe-managed-rule-group`.

### AWS CLI

Per recuperare la descrizione di un gruppo di regole gestito

Di seguito viene `describe-managed-rule-group` recuperata la descrizione di un gruppo di regole AWS gestito.

```
aws wafv2 describe-managed-rule-group \  
  --vendor-name AWS \  
  --name AWSManagedRulesCommonRuleSet \  
  --scope REGIONAL
```

Output:

```
{  
  "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": "EC2Meta-data-SSRF_BODY",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "EC2Meta-data-SSRF_COOKIE",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "EC2Meta-data-SSRF_URI_PATH",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "EC2Meta-data-SSRF_QUERY_ARGUMENTS",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "GenericLFI_QUERY_ARGUMENTS",
    "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_URIPATH",
  "Action": {
    "Block": {}
  }
},
{
```

```
    "Name": "CrossSiteScripting_COOKIE",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "CrossSiteScripting_QUERYARGUMENTS",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "CrossSiteScripting_BODY",
    "Action": {
      "Block": {}
    }
  },
  {
    "Name": "CrossSiteScripting_URI_PATH",
    "Action": {
      "Block": {}
    }
  }
]
}
```

Per ulteriori informazioni, consulta [Managed Rule Groups](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [DescribeManagedRuleGroup](#) in AWS CLI Command Reference.

## disassociate-web-acl

Il seguente esempio di codice mostra come utilizzare `disassociate-web-acl`.

### AWS CLI

Per dissociare un Web ACL da una risorsa regionale AWS

L'esempio seguente rimuove qualsiasi ACL associazione Web esistente dall'Application Load Balancer specificato.

```
aws wafv2 disassociate-web-acl \
```

```
--resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/  
app/waf-cli-alb/1ea17125f8b25a2a \  
--region us-west-2
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Associare o dissociare un Web ACL con una AWS risorsa](#) nella AWS Firewall AWS WAF Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [DisassociateWebAcl](#) in AWS CLI Command Reference.

## get-ip-set

Il seguente esempio di codice mostra come utilizzare `get-ip-set`.

### AWS CLI

Per recuperare un set IP specifico

Quanto segue `get-ip-set` recupera il set IP con il nome, l'ambito e l'ID specificati. È possibile ottenere l'ID per un set IP dai comandi `create-ip-set` e `list-ip-sets`.

```
aws wafv2 get-ip-set \  
  --name testip \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE1111
```

Output:

```
{  
  "IPSet": {  
    "Description": "",  
    "Name": "testip",  
    "IPAddressVersion": "IPV4",  
    "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE1111",  
    "ARN": "arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE1111",  
    "Addresses": [  
      "192.0.2.0/16"  
    ]  
  },  
  "LockToken": "447e55ac-2396-4c6d-b9f9-86b67c17f8b5"
```

```
}
```

Per ulteriori informazioni, vedere [IP Sets and Regex Pattern Sets](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [GetIpSet](#) in AWS CLI Command Reference.

## get-logging-configuration

Il seguente esempio di codice mostra come utilizzare `get-logging-configuration`.

### AWS CLI

Per recuperare le configurazioni di registrazione per un Web ACL

Quanto segue `get-logging-configuration` recupera la configurazione di registrazione per il Web specificato. ACL

```
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
```

### Output:

```
{
  "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"
    ]
  }
}
```

Per ulteriori informazioni, vedere [Logging Web ACL Traffic Information](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [GetLoggingConfiguration](#) in AWS CLI Command Reference.

## get-rate-based-statement-managed-keys

Il seguente esempio di codice mostra come utilizzare `get-rate-based-statement-managed-keys`.

### AWS CLI

Per recuperare un elenco di indirizzi IP bloccati da una regola basata sulla velocità

Di seguito `get-rate-based-statement-managed-keys` vengono recuperati gli indirizzi IP attualmente bloccati da una regola basata sulla tariffa utilizzata per un'applicazione regionale.

```
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
```

### Output:

```
{
  "ManagedKeysIPV4": {
    "IPAddressVersion": "IPV4",
    "Addresses": [
      "198.51.100.0/32"
    ]
  },
  "ManagedKeysIPV6": {
    "IPAddressVersion": "IPV6",
    "Addresses": [
    ]
  }
}
```

Per ulteriori informazioni, vedere [Rate-Based Rule Statement](#) nella AWS Firewall Manager and AWS WAF AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [GetRateBasedStatementManagedKeys](#) in AWS CLI Command Reference.

## get-regex-pattern-set

Il seguente esempio di codice mostra come utilizzare `get-regex-pattern-set`.

### AWS CLI

Per recuperare un set di pattern regex specifico

Quanto segue `get-regex-pattern-set` recupera il set di pattern regex con il nome, l'ambito, la regione e l'ID specificati. È possibile ottenere l'ID per un set di pattern regex dai comandi `create-regex-pattern-set` e `list-regex-pattern-sets`

```
aws wafv2 get-regex-pattern-set \
  --name regexPatterSet01 \
  --scope REGIONAL \
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --region us-west-2
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, vedere [IP Sets and Regex Pattern Sets](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [GetRegexPatternSet](#) in AWS CLI Command Reference.

## get-rule-group

Il seguente esempio di codice mostra come utilizzare `get-rule-group`.

### AWS CLI

Per recuperare uno specifico gruppo di regole personalizzate

Quanto segue `get-rule-group` recupera il gruppo di regole personalizzato con il nome, l'ambito e l'ID specificati. È possibile ottenere l'ID per un gruppo di regole dai comandi `create-rule-group` e `list-rule-groups`.

```
aws wafv2 get-rule-group \  
  --name ff \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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"
}

```

Per ulteriori informazioni, vedere [Managing Your Own Rule Group](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [GetRuleGroup](#) in AWS CLI Command Reference.

## get-sampled-requests

Il seguente esempio di codice mostra come utilizzare `get-sampled-requests`.

### AWS CLI

Per recuperare un esempio di richieste Web per un sito Web ACL

Quanto segue `get-sampled-requests` recupera le richieste Web campionate per il WebACL, la metrica della regola e l'intervallo di tempo specificati.

```
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
```

Output:

```
{  
  "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
}
```

Per ulteriori informazioni, vedere [Visualizzazione di un esempio di richieste Web](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [GetSampledRequests](#) in AWS CLI Command Reference.

## get-web-acl-for-resource

Il seguente esempio di codice mostra come utilizzare `get-web-acl-for-resource`.

## AWS CLI

Per recuperare il Web ACL associato a una risorsa AWS

Quanto segue `get-web-acl-for-resource` recupera il JSON file per ACL il Web associato alla risorsa specificata.

```
aws wafv2 get-web-acl-for-resource \
  --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/
app/waf-cli-alb/1ea17125f8b25a2a
```

Output:

```
{
  "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"

```

```
}  
}
```

Per ulteriori informazioni, consulta [Associare o dissociare un Web ACL con una AWS risorsa](#) nella AWS Firewall AWS WAF Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [GetWebAclForResource](#) in AWS CLI Command Reference.

## get-web-acl

Il seguente esempio di codice mostra come utilizzare `get-web-acl`.

### AWS CLI

Per recuperare un web ACL

Quanto segue `get-web-acl` recupera il Web ACL con il nome, l'ambito e l'ID specificati. È possibile ottenere l'ID di un Web ACL dai comandi `create-web-acl` e `list-web-acls`.

```
aws wafv2 get-web-acl \  
  --name test01 \  
  --scope REGIONAL \  
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Output:

```
{  
  "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"
}
```

Per ulteriori informazioni, vedere [Managing and Using a Web Access Control List \(WebACL\)](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [GetWebAcl](#) in AWS CLI Command Reference.

## list-available-managed-rule-groups

Il seguente esempio di codice mostra come utilizzare `list-available-managed-rule-groups`.

### AWS CLI

Per recuperare i gruppi di regole gestiti

Quanto segue `list-available-managed-rule-groups` restituisce l'elenco di tutti i gruppi di regole gestiti attualmente disponibili per l'uso nei WebACLs.

```
aws wafv2 list-available-managed-rule-groups \
  --scope REGIONAL
```

Output:

```
{
  "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."
}
]
```

Per ulteriori informazioni, consulta [Managed Rule Groups](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [ListAvailableManagedRuleGroups](#) in AWS CLI Command Reference.

## list-ip-sets

Il seguente esempio di codice mostra come utilizzare `list-ip-sets`.

## AWS CLI

Per recuperare un elenco di set IP

Quanto segue `list-ip-sets` recupera tutti i set IP per l'account che hanno un ambito regionale.

```
aws wafv2 list-ip-sets \  
  --scope REGIONAL
```

Output:

```
{  
  "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"  
}
```

Per ulteriori informazioni, vedere [IP Sets and Regex Pattern Sets](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [ListIpSets](#) in AWS CLI Command Reference.

## list-logging-configurations

Il seguente esempio di codice mostra come utilizzare `list-logging-configurations`.

### AWS CLI

Per recuperare un elenco di tutte le configurazioni di registrazione per una regione

Di seguito vengono `list-logging-configurations` recuperate tutte le configurazioni di registrazione per il Web destinate all'ACLsuso regionale nella regione. `us-west-2`

```
aws wafv2 list-logging-configurations \  
  --scope REGIONAL \  
  --region us-west-2
```

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

Output:

```
{
  "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"
      ]
    }
  ]
}
```

Per ulteriori informazioni, vedere [Logging Web ACL Traffic Information](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [ListLoggingConfigurations](#) in AWS CLI Command Reference.

## list-regex-pattern-sets

Il seguente esempio di codice mostra come utilizzare `list-regex-pattern-sets`.

### AWS CLI

Per recuperare un elenco di set di pattern regex

Quanto segue `list-regex-pattern-sets` recupera tutti i set di pattern regex per l'account definiti nella regione. `us-west-2`

```
aws wafv2 list-regex-pattern-sets \  
--scope REGIONAL \  
--region us-west-2
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [IP Sets and Regex Pattern Sets](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [ListRegexPatternSets](#) in AWS CLI Command Reference.

## list-resources-for-web-acl

Il seguente esempio di codice mostra come utilizzare `list-resources-for-web-acl`.

### AWS CLI

Per recuperare le risorse associate a un Web ACL

Quando segue `list-resources-for-web-acl` recupera REST API le risorse del API Gateway attualmente associate al Web specificato ACL nella regione. `us-west-2`

```
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
```

Output:

```
{
  "ResourceArns": [
    "arn:aws:apigateway:us-west-2::/restapis/EXAMPLE111/stages/testing"
  ]
}
```

Per ulteriori informazioni, consulta [Associare o dissociare un Web ACL con una AWS risorsa](#) nella AWS Firewall AWS WAF Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [ListResourcesForWebAcl](#) in AWS CLI Command Reference.

## list-rule-groups

Il seguente esempio di codice mostra come utilizzare `list-rule-groups`.

AWS CLI

Per recuperare un elenco di gruppi di regole personalizzati

Quando segue `list-rule-groups` recupera tutti i gruppi di regole personalizzati definiti per l'account per l'ambito e la posizione della regione specificati.

```
aws wafv2 list-rule-groups \
  --scope REGIONAL \
  --region us-west-2
```

Output:

```
{
  "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"
}

```

Per ulteriori informazioni, vedere [Managing Your Own Rule Group](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [ListRuleGroups](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per recuperare tutti i tag di una risorsa AWS WAF

Quanto segue `list-tags-for-resource` recupera l'elenco di tutte le coppie di tag, chiavi e valori per il web specificato. ACL

```

aws wafv2 list-tags-for-resource \
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/testwebacl2/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

```

Output:

```
{
```

```

    "NextMarker": "",
    "TagInfoForResource": {
      "ResourceARN": "arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
testwebacl2/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "TagList": [
        ]
      }
    }
  }
}

```

Per ulteriori informazioni, consulta la [Guida introduttiva AWS WAF](#) nella Guida per sviluppatori di AWS Firewall Manager and AWS Shield Advanced Developer Guide.AWS WAF

- Per API i dettagli, vedere [ListTagsForResource](#) in AWS CLI Command Reference.

## list-web-acls

Il seguente esempio di codice mostra come utilizzare `list-web-acls`.

### AWS CLI

Per recuperare il Web da un ACLs cannocchiale

Quanto segue `list-web-acls` recupera tutti ACLs i Web definiti per l'account per l'ambito specificato.

```

aws wafv2 list-web-acls \
  --scope REGIONAL

```

Output:

```

{
  "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"
    }
  ]
}

```

```
]
}
```

Per ulteriori informazioni, vedere [Managing and Using a Web Access Control List \(WebACL\)](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [ListWebAcls](#) in AWS CLI Command Reference.

## put-logging-configuration

Il seguente esempio di codice mostra come utilizzare `put-logging-configuration`.

### AWS CLI

Per aggiungere una configurazione di registrazione a un sito Web ACL

Quanto segue `put-logging-configuration` aggiunge la configurazione di `aws-waf-logs-custom-transformation` registrazione di Amazon Kinesis Data Firehose al Web ACL specificato, senza che i campi vengano eliminati dai log.

```
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
```

Output:

```
{
  "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"
    ]
  }
}
```

Per ulteriori informazioni, vedere [Logging Web ACL Traffic Information](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [PutLoggingConfiguration](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

### AWS CLI

Per aggiungere tag a una AWS WAF risorsa

L'`tag-resource` esempio seguente aggiunge un tag con una chiave Name e un valore impostato su AWS WAF al Web specificato ACL.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la [Guida introduttiva AWS WAF](#) nella Guida per sviluppatori di AWS Firewall Manager and AWS Shield Advanced Developer Guide. AWS WAF

- Per API i dettagli, vedere [TagResource](#) in AWS CLI Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare `untag-resource`.

### AWS CLI

Per rimuovere i tag da una AWS WAF risorsa

L'`untag-resource` esempio seguente rimuove il tag con la chiave KeyName dal Web specificato ACL.

```
aws wafv2 untag-resource \  
  --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/  
apiGatewayWebAcl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \  
  --tag-keys "KeyName"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la [Guida introduttiva AWS WAF](#) nella Guida per sviluppatori di AWS Firewall Manager and AWS Shield Advanced Developer Guide.AWS WAF

- Per API i dettagli, vedere [UntagResource](#) in AWS CLI Command Reference.

## update-ip-set

Il seguente esempio di codice mostra come utilizzare `update-ip-set`.

### AWS CLI

Per modificare le impostazioni di un set IP esistente

Quanto segue `update-ip-set` aggiorna le impostazioni per il set IP specificato. Questa chiamata richiede un ID, che è possibile ottenere dalla chiamata `list-ip-sets`, e un token di blocco che è possibile ottenere dalle chiamate, `list-ip-sets` e `get-ip-set`. Questa chiamata restituisce anche un token di blocco che è possibile utilizzare per un aggiornamento successivo.

```
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
```

Output:

```
{  
  "NextLockToken": "0674c84b-0304-47fe-8728-c6bff46af8fc"  
}
```

Per ulteriori informazioni, vedere [IP Sets and Regex Pattern Sets](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [UpdateIpSet](#) in AWS CLI Command Reference.

## update-regex-pattern-set

Il seguente esempio di codice mostra come utilizzare `update-regex-pattern-set`.

## AWS CLI

Per modificare le impostazioni per un set di pattern regex esistente

Quando segue `update-regex-pattern-set` aggiorna le impostazioni per il set di pattern regex specificato. Questa chiamata richiede un ID, che è possibile ottenere dalla chiamata `list-regex-pattern-sets`, e un token di blocco che è possibile ottenere dalle chiamate, `list-regex-pattern-sets` e `get-regex-pattern-set`. Questa chiamata restituisce anche un token di blocco che è possibile utilizzare per un aggiornamento successivo.

```
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
```

Output:

```
{
  "NextLockToken": "12ebc73e-fa68-417d-a9b8-2bdd761a4fa5"
}
```

Per ulteriori informazioni, vedere [IP Sets and Regex Pattern Sets](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [UpdateRegexPatternSet](#) in AWS CLI Command Reference.

## update-rule-group

Il seguente esempio di codice mostra come utilizzare `update-rule-group`.

## AWS CLI

Per aggiornare un gruppo di regole personalizzato

Quando segue `update-rule-group` modifica la configurazione della visibilità per un gruppo di regole personalizzato esistente. Questa chiamata richiede un ID, che è possibile ottenere dalla chiamata `list-rule-groups`, e un token di blocco che è possibile ottenere dalle chiamate, `list-rule-groups` e `get-rule-group`. Questa chiamata restituisce anche un token di blocco che è possibile utilizzare per un aggiornamento successivo.

```
aws wafv2 update-rule-group \
  --name TestRuleGroup \
  --scope REGIONAL \
  --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
  --lock-token 7b3bcec2-0000-0000-0000-563bf47249f0 \
  --visibility-
config SampledRequestsEnabled=false,CloudWatchMetricsEnabled=false,MetricName=TestMetricsFor
\
  --region us-west-2
```

Output:

```
{
  "NextLockToken": "1eb5ec48-0000-0000-0000-ee9b906c541e"
}
```

Per ulteriori informazioni, vedere [Managing Your Own Rule Group](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [UpdateRuleGroup](#) in AWS CLI Command Reference.

## update-web-acl

Il seguente esempio di codice mostra come utilizzare `update-web-acl`.

### AWS CLI

Per aggiornare un sito web ACL

Quando segue `update-web-acl` modifica le impostazioni per un Web esistente ACL. Questa chiamata richiede un ID, che è possibile ottenere dalla chiamata `list-web-acls`, un token di blocco e altre impostazioni, che è possibile ottenere dalla chiamata `get-web-acl`. Questa chiamata restituisce anche un token di blocco che è possibile utilizzare per un aggiornamento successivo.

```
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=NewMetricTest  
\   
--rules file://waf-rule.json \  
--region us-west-2
```

Output:

```
{  
  "NextLockToken": "714a0cfb-0000-0000-0000-2959c8b9a684"  
}
```

Per ulteriori informazioni, vedere [Managing and Using a Web Access Control List \(WebACL\)](#) nella AWS WAF AWS Firewall Manager and AWS Shield Advanced Developer Guide.

- Per API i dettagli, vedere [UpdateWebAcl](#) in AWS CLI Command Reference.

## WorkDocs Esempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con Amazon WorkDocs.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **abort-document-version-upload**

Il seguente esempio di codice mostra come utilizzare `abort-document-version-upload`.

AWS CLI

Per interrompere il caricamento della versione di un documento



Questo esempio interrompe il caricamento di una versione del documento avviato in precedenza.

Comando:

```
aws workdocs abort-document-version-upload --document-  
id feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f01267313209835f2 --version-  
id 1536773972914-ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dccccd417da9313
```

Output:

```
None
```

- Per API i dettagli, vedere [AbortDocumentVersionUpload](#) in AWS CLI Command Reference.

## activate-user

Il seguente esempio di codice mostra come utilizzare `activate-user`.

### AWS CLI

Per attivare un utente

Questo esempio attiva un utente inattivo.

Comando:

```
aws workdocs activate-user --user-  
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

Output:

```
{  
  "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"
      }
    }
  }
}

```

- Per API i dettagli, vedere [ActivateUser](#) in AWS CLI Command Reference.

## add-resource-permissions

Il seguente esempio di codice mostra come utilizzare `add-resource-permissions`.

### AWS CLI

Per aggiungere le autorizzazioni per una risorsa

Questo esempio aggiunge le autorizzazioni alla risorsa per i principali specificati.

Comando:

```

aws workdocs add-resource-permissions --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
principals Id=anonymous, Type=ANONYMOUS, Role=VIEWER

```

Output:

```

{
  "ShareResults": [
    {
      "PrincipalId": "anonymous",
      "Role": "VIEWER",
      "Status": "SUCCESS",
    }
  ]
}

```

```

    "ShareId":
      "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
      "StatusMessage": ""
    }
  ]
}

```

- Per API i dettagli, vedere [AddResourcePermissions](#) in AWS CLI Command Reference.

## create-comment

Il seguente esempio di codice mostra come utilizzare `create-comment`.

### AWS CLI

Per aggiungere un nuovo commento

Questo esempio aggiunge un nuovo commento alla versione del documento specificata.

Comando:

```

aws workdocs create-comment --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
text "This is a comment."

```

Output:

```

{
  "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"  
}  
}
```

- Per API i dettagli, vedere [CreateComment](#) in AWS CLI Command Reference.

## create-custom-metadata

Il seguente esempio di codice mostra come utilizzare `create-custom-metadata`.

### AWS CLI

Per creare metadati personalizzati

Questo esempio crea metadati personalizzati per il documento specificato.

Comando:

```
aws workdocs create-custom-metadata --resource-  
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --custom-  
metadata KeyName1=example,KeyName2=example2
```

Output:

```
None
```

- Per API i dettagli, vedere [CreateCustomMetadata](#) in AWS CLI Command Reference.

## create-folder

Il seguente esempio di codice mostra come utilizzare `create-folder`.

### AWS CLI

Per creare una cartella

Questo esempio crea una cartella.

Comando:

```
aws workdocs create-folder --name documents --parent-folder-  
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

Output:

```
{  
  "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  
  }  
}
```

- Per API i dettagli, vedere [CreateFolder](#) in AWS CLI Command Reference.

## create-labels

Il seguente esempio di codice mostra come utilizzare `create-labels`.

AWS CLI

Per creare etichette

Questo esempio crea una serie di etichette per un documento.

Comando:

```
aws workdocs create-labels --resource-  
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --  
labels "documents" "examples" "my_documents"
```

Output:

None

- Per API i dettagli, vedere [CreateLabels](#) in AWS CLI Command Reference.

## create-notification-subscription

Il seguente esempio di codice mostra come utilizzare `create-notification-subscription`.

### AWS CLI

Per creare una sottoscrizione alle notifiche

L'`create-notification-subscription` esempio seguente configura un abbonamento alle notifiche per l' `WorkDocs` organizzazione Amazon specificata.

```
aws workdocs create-notification-subscription \  
  --organization-id d-123456789c \  
  --protocol HTTPS \  
  --subscription-type ALL \  
  --notification-endpoint "https://example.com/example"
```

Output:

```
{  
  "Subscription": {  
    "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",  
    "EndPoint": "https://example.com/example",  
    "Protocol": "HTTPS"  
  }  
}
```

Per ulteriori informazioni, consulta [Abbonarsi alle notifiche](#) nell'Amazon WorkDocs Developer Guide.

- Per API i dettagli, consulta [CreateNotificationSubscription AWS CLI](#) Command Reference.

## create-user

Il seguente esempio di codice mostra come utilizzare `create-user`.

## AWS CLI

Per creare un nuovo utente

Questo esempio crea un nuovo utente in una directory Simple AD o Microsoft AD.

Comando:

```
aws workdocs create-user --organization-id d-926726012c --username exampleUser2
--email-address exampleUser2@site.awsapps.com --given-name example2Name --
surname example2Surname --password examplePa$$w0rd
```

Output:

```
{
  "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"
      }
    }
  }
}
```

- Per API i dettagli, vedere [CreateUser](#) in AWS CLI Command Reference.

## deactivate-user

Il seguente esempio di codice mostra come utilizzare `deactivate-user`.

### AWS CLI

Per disattivare un utente

Questo esempio disattiva un utente attivo.

Comando:

```
aws workdocs deactivate-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

Output:

```
None
```

- Per API i dettagli, vedere [DeactivateUser](#) in AWS CLI Command Reference.

## delete-comment

Il seguente esempio di codice mostra come utilizzare `delete-comment`.

### AWS CLI

Per eliminare un commento specificato da una versione del documento

Questo esempio elimina il commento specificato dalla versione del documento specificata.

Comando:

```
aws workdocs delete-comment --document-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --comment-id 1534799058197-c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5
```

Output:

```
None
```



- Per API i dettagli, vedere [DeleteComment](#) in AWS CLI Command Reference.

## **delete-custom-metadata**

Il seguente esempio di codice mostra come utilizzare `delete-custom-metadata`.

### AWS CLI

Per eliminare i metadati personalizzati da una risorsa

Questo esempio elimina tutti i metadati personalizzati dalla risorsa specificata.

Comando:

```
aws workdocs delete-custom-metadata --resource-  
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --delete-all
```

Output:

```
None
```

- Per API i dettagli, vedere [DeleteCustomMetadata](#) in AWS CLI Command Reference.

## **delete-document**

Il seguente esempio di codice mostra come utilizzare `delete-document`.

### AWS CLI

Per eliminare un documento

Questo esempio elimina il documento specificato.

Comando:

```
aws workdocs delete-document --document-  
id b83ed5e5b167b65ef69de9d597627ff1a0d4f07a45e67f1fab7d26b54427de0a
```

Output:

```
None
```

- Per API i dettagli, vedere [DeleteDocument](#) in AWS CLI Command Reference.

## delete-folder-contents

Il seguente esempio di codice mostra come utilizzare `delete-folder-contents`.

### AWS CLI

Per eliminare il contenuto di una cartella

Questo esempio elimina il contenuto della cartella specificata.

Comando:

```
aws workdocs delete-folder-contents --folder-  
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

Output:

```
None
```

- Per API i dettagli, vedere [DeleteFolderContents](#) in AWS CLI Command Reference.

## delete-folder

Il seguente esempio di codice mostra come utilizzare `delete-folder`.

### AWS CLI

Per eliminare una cartella

Questo esempio elimina la cartella specificata.

Comando:

```
aws workdocs delete-folder --folder-  
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

Output:

```
None
```

- Per API i dettagli, vedere [DeleteFolder](#) in AWS CLI Command Reference.

## delete-labels

Il seguente esempio di codice mostra come utilizzare `delete-labels`.

AWS CLI

Per eliminare le etichette

Questo esempio elimina le etichette specificate da un documento.

Comando:

```
aws workdocs delete-labels --resource-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --labels "documents" "examples"
```

Output:

```
None
```

- Per API i dettagli, vedere [DeleteLabels](#) in AWS CLI Command Reference.

## delete-notification-subscription

Il seguente esempio di codice mostra come utilizzare `delete-notification-subscription`.

AWS CLI

Per eliminare una sottoscrizione alle notifiche

L'esempio seguente elimina la sottoscrizione di notifica specificata.

```
aws workdocs delete-notification-subscription \
  --subscription-id 123ab4c5-678d-901e-f23g-45h6789j0123 \
  --organization-id d-123456789c
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Abbonarsi alle notifiche](#) nell'Amazon WorkDocs Developer Guide.

- Per API i dettagli, consulta [DeleteNotificationSubscription AWS CLI Command Reference](#).

## **delete-user**

Il seguente esempio di codice mostra come utilizzare `delete-user`.

### AWS CLI

Per eliminare un utente

Questo esempio elimina un utente.

Comando:

```
aws workdocs delete-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

Output:

```
None
```

- Per API i dettagli, vedere [DeleteUser](#) in AWS CLI Command Reference.

## **describe-activities**

Il seguente esempio di codice mostra come utilizzare `describe-activities`.

### AWS CLI

Per ottenere un elenco delle attività degli utenti

Questo esempio restituisce un elenco delle attività utente più recenti per l'organizzazione specificata, con un limite impostato per le ultime due attività.

Comando:

```
aws workdocs describe-activities --organization-id d-926726012c --limit 2
```

## Output:

```
{
  "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":
    "DnF1ZXJ5VGh1bkZldGNoAgAAAAAAS7Fm1TaU10d1FTU1h1UU00VVFibD1RWHcAAAAAAAJTRY3bWh5eUgzaVF1ZX
  }

```

- Per API i dettagli, vedere [DescribeActivities](#) in AWS CLI Command Reference.

## describe-comments

Il seguente esempio di codice mostra come utilizzare `describe-comments`.

### AWS CLI

Per elencare tutti i commenti per una versione del documento specificata

Questo esempio elenca tutti i commenti per la versione del documento specificata.

Comando:

```

aws workdocs describe-comments --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920

```

Output:

```

{
  "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"
    }
  ]
}

```

```
}
```

- Per API i dettagli, vedere [DescribeComments](#) in AWS CLI Command Reference.

## describe-document-versions

Il seguente esempio di codice mostra come utilizzare `describe-document-versions`.

### AWS CLI

Per recuperare le versioni di un documento

Questo esempio recupera le versioni del documento specificato, incluse le versioni inizializzate e una URL per il documento di origine.

Comando:

```
aws workdocs describe-document-versions --document-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields SOURCE
```

Output:

```
{
  "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"
    }
  }
]
}

```

- Per API i dettagli, vedere [DescribeDocumentVersions](#) in AWS CLI Command Reference.

## describe-folder-contents

Il seguente esempio di codice mostra come utilizzare `describe-folder-contents`.

### AWS CLI

Per descrivere il contenuto di una cartella



Questo esempio descrive tutti i contenuti attivi della cartella specificata, inclusi i documenti e le sottocartelle, ordinati per data in ordine crescente.

Comando:

```
aws workdocs describe-folder-contents --folder-id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --sort DATE --order ASCENDING --type ALL
```

Output:

```
{
  "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"
}
]
}

```

- Per API i dettagli, vedere [DescribeFolderContents](#) in AWS CLI Command Reference.

## describe-groups

Il seguente esempio di codice mostra come utilizzare `describe-groups`.

### AWS CLI

Per recuperare un elenco di gruppi

L'esempio seguente elenca i gruppi associati all'organizzazione Amazon specificata.

```

aws workdocs describe-groups \
  --search-query "e" \
  --organization-id d-123456789c

```

Output:

```

{
  "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"
    }
  ]
}

```

```
]
}
```

Per ulteriori informazioni, consulta la sezione [Getting Started with Amazon WorkDocs](#) nella Amazon WorkDocs Administration Guide.

- Per API i dettagli, consulta [DescribeGroups AWS CLI Command Reference](#).

## describe-notification-subscriptions

Il seguente esempio di codice mostra come utilizzare `describe-notification-subscriptions`.

### AWS CLI

Per recuperare un elenco di sottoscrizioni alle notifiche

L'`describe-notification-subscriptions`esempio seguente recupera gli abbonamenti alle notifiche per l'organizzazione Amazon WorkDocs specificata.

```
aws workdocs describe-notification-subscriptions \
  --organization-id d-123456789c
```

Output:

```
{
  "Subscriptions": [
    {
      "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",
      "EndPoint": "https://example.com/example",
      "Protocol": "HTTPS"
    }
  ]
}
```

Per ulteriori informazioni, consulta [Abbonarsi alle notifiche](#) nell'Amazon WorkDocs Developer Guide.

- Per API i dettagli, consulta [DescribeNotificationSubscriptions AWS CLI Command Reference](#).

## describe-resource-permissions

Il seguente esempio di codice mostra come utilizzare `describe-resource-permissions`.

## AWS CLI

Per ottenere un elenco di autorizzazioni per una risorsa

Questo esempio restituisce un elenco delle autorizzazioni per la risorsa specificata (documento o cartella).

Comando:

```
aws workdocs describe-resource-permissions --resource-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3
```

Output:

```
{
  "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"
        }
      ]
    }
  ]
}
```

```

    }
  ]
}

```

- Per API i dettagli, vedere [DescribeResourcePermissions](#) in AWS CLI Command Reference.

## describe-users

Il seguente esempio di codice mostra come utilizzare `describe-users`.

### AWS CLI

Per recuperare i dettagli per utenti specifici

Questo esempio recupera i dettagli per tutti gli utenti dell'organizzazione specificata.

Comando:

```
aws workdocs describe-users --organization-id d-926726012c
```

Output:

```

{
  "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
  }
]
}

```

- Per API i dettagli, vedere [DescribeUsers](#) in AWS CLI Command Reference.

## get-document-path

Il seguente esempio di codice mostra come utilizzare `get-document-path`.

### AWS CLI

Per recuperare le informazioni sul percorso di un documento

Questo esempio recupera le informazioni sul percorso (gerarchia dalla cartella principale) per il documento specificato e include i nomi delle cartelle principali.

Comando:

```

aws workdocs get-document-path --document-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields NAME

```

Output:

```

{
  "Path": {
    "Components": [
      {
        "Id":
"a43d29cbb8e7c4d25cfce8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",

```

```

        "Name": "/"
      },
      {
        "Id":
"1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
        "Name": "Top Level Folder"
      },
      {
        "Id":
"d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
        "Name": "exampleDoc.docx"
      }
    ]
  }
}

```

- Per API i dettagli, vedere [GetDocumentPath](#) in AWS CLI Command Reference.

## get-document-version

Il seguente esempio di codice mostra come utilizzare `get-document-version`.

### AWS CLI

Per recuperare i metadati della versione per un documento specificato

Questo esempio recupera i metadati della versione per il documento specificato, tra cui un'origine URL e i metadati personalizzati.

Comando:

```

aws workdocs get-document-version --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
fields SOURCE --include-custom-metadata

```

Output:

```

{
  "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"
    }
  }
}

```

- Per API i dettagli, vedete [GetDocumentVersion](#) in AWS CLI Command Reference.

## get-document

Il seguente esempio di codice mostra come utilizzare `get-document`.

### AWS CLI

Per recuperare i dettagli del documento

Questo esempio recupera i dettagli del documento specificato.

Comando:

```
aws workdocs get-document --document-id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65
```

Output:

```
{
  "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"
  }
}

```

- Per API i dettagli, vedere [GetDocument](#) in AWS CLI Command Reference.

## get-folder-path

Il seguente esempio di codice mostra come utilizzare `get-folder-path`.

### AWS CLI

Per recuperare le informazioni sul percorso di una cartella

Questo esempio recupera le informazioni sul percorso (gerarchia dalla cartella principale) per la cartella specificata e include i nomi delle cartelle principali.

Comando:

```
aws workdocs get-folder-path --folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --fields NAME
```

Output:

```
{
  "Path": {
    "Components": [
      {
        "Id":
        "a43d29cbb8e7c4d25cfee8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",
        "Name": "/"
      },
      {
        "Id":
        "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
        "Name": "Top Level Folder"
      },
      {
        "Id":
        "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
        "Name": "Sublevel Folder"
      }
    ]
  }
}
```

- Per API i dettagli, vedere [GetFolderPath](#) in AWS CLI Command Reference.

## get-folder

Il seguente esempio di codice mostra come utilizzare `get-folder`.

### AWS CLI

Per recuperare i metadati di una cartella

Questo esempio recupera i metadati per la cartella specificata.

Comando:

```
aws workdocs get-folder --folder-id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

Output:

```
{
```

```
"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
}
```

- Per API i dettagli, vedere [GetFolder](#) in AWS CLI Command Reference.

## get-resources

Il seguente esempio di codice mostra come utilizzare `get-resources`.

### AWS CLI

Per recuperare risorse condivise

L'`get-resources` esempio seguente recupera le risorse condivise con l' WorkDocs utente Amazon specificato.

```
aws workdocs get-resources \
  --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333" \
  --collection-type SHARED_WITH_ME
```

Output:

```
{
  "Folders": [],
  "Documents": []
}
```

Per ulteriori informazioni, consulta [Sharing Files and Folders](#) nella Amazon WorkDocs User Guide.

- Per API i dettagli, consulta [GetResources AWS CLI Command Reference](#).

## initiate-document-version-upload

Il seguente esempio di codice mostra come utilizzare `initiate-document-version-upload`.

### AWS CLI

Per avviare il caricamento di una versione del documento

L'`initiate-document-upload`esempio seguente crea un nuovo oggetto documento e un nuovo oggetto versione.

```
aws workdocs initiate-document-version-upload \  
  --name exampledocname \  
  --parent-folder-  
id eacd546d952531c633452ed67cac23161aa0d5df2e8061223a59e8f67e7b6189
```

Output:

```
{  
  "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"  
  }  
}
```

- Per API i dettagli, vedere [InitiateDocumentVersionUpload](#) in AWS CLI Command Reference.

## remove-all-resource-permissions

Il seguente esempio di codice mostra come utilizzare `remove-all-resource-permissions`.

### AWS CLI

Per rimuovere tutte le autorizzazioni da una risorsa specificata

Questo esempio rimuove tutte le autorizzazioni dalla risorsa specificata.

Comando:

```
aws workdocs remove-all-resource-permissions --resource-  
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

Output:

```
None
```

- Per API i dettagli, vedere [RemoveAllResourcePermissions](#) in AWS CLI Command Reference.

## remove-resource-permission

Il seguente esempio di codice mostra come utilizzare `remove-resource-permission`.

## AWS CLI

Per rimuovere le autorizzazioni da una risorsa

Questo esempio rimuove le autorizzazioni dalla risorsa per il principale specificato.

Comando:

```
aws workdocs remove-resource-permission --resource-id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --principal-id anonymous
```

Output:

```
None
```

- Per API i dettagli, vedere [RemoveResourcePermission](#) in AWS CLI Command Reference.

## update-document-version

Il seguente esempio di codice mostra come utilizzare `update-document-version`.

## AWS CLI

Per modificare lo stato della versione di un documento in Attivo

In questo esempio lo stato della versione del documento viene impostato su Attivo.

Comando:

```
aws workdocs update-document-version --document-id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --version-status ACTIVE
```

Output:

```
None
```

- Per API i dettagli, vedere [UpdateDocumentVersion](#) in AWS CLI Command Reference.

## update-document

Il seguente esempio di codice mostra come utilizzare `update-document`.

### AWS CLI

Per aggiornare un documento

Questo esempio aggiorna il nome e la cartella principale di un documento.

Comando:

```
aws workdocs update-document --document-  
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --  
name updatedDoc --parent-folder-  
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

Output:

```
None
```

- Per API i dettagli, vedere [UpdateDocument](#) in AWS CLI Command Reference.

## update-folder

Il seguente esempio di codice mostra come utilizzare `update-folder`.

### AWS CLI

Per aggiornare una cartella

Questo esempio aggiorna il nome e la cartella principale di una cartella.

Comando:

```
aws workdocs update-folder --folder-  
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --  
name exampleFolder1 --parent-folder-  
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

Output:

None

- Per API i dettagli, vedere [UpdateFolder](#) in AWS CLI Command Reference.

## update-user

Il seguente esempio di codice mostra come utilizzare `update-user`.

### AWS CLI

Per aggiornare un utente

Questo esempio aggiorna il fuso orario per l'utente specificato.

Comando:

```
aws workdocs update-user --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c" --time-zone-id "America/Los_Angeles"
```

Output:

```
{
  "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"  
  }  
}  
}
```

- Per API i dettagli, vedere [UpdateUser](#) in AWS CLI Command Reference.

## WorkMail Esempi di utilizzo di Amazon AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con Amazon WorkMail.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, dove puoi trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **associate-delegate-to-resource**

Il seguente esempio di codice mostra come utilizzare `associate-delegate-to-resource`.

#### AWS CLI

Per aggiungere un delegato a una risorsa

Il `associate-delegate-to-resource` comando seguente aggiunge un delegato a una risorsa.

```
aws workmail associate-delegate-to-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \  
  --role-arn arn:aws:iam::123456789012:role/WorkMailAdmin
```

```
--entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [AssociateDelegateToResource](#) in AWS CLI Command Reference.

## associate-member-to-group

Il seguente esempio di codice mostra come utilizzare `associate-member-to-group`.

### AWS CLI

Per aggiungere un membro a un gruppo

Il `associate-member-to-group` comando seguente aggiunge il membro specificato a un gruppo.

```
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
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [AssociateMemberToGroup](#) in AWS CLI Command Reference.

## create-alias

Il seguente esempio di codice mostra come utilizzare `create-alias`.

### AWS CLI

Per creare un alias

Il `create-alias` comando seguente crea un alias per l'entità specificata (utente o gruppo).

```
aws workmail create-alias \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --alias exampleAlias@site.awsapps.com
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [CreateAlias](#) in AWS CLI Command Reference.

## create-group

Il seguente esempio di codice mostra come utilizzare `create-group`.

### AWS CLI

Per creare un nuovo gruppo

Il `create-group` comando seguente crea un nuovo gruppo per l'organizzazione specificata.

```
aws workmail create-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --name exampleGroup1
```

Output:

```
{  
  "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444"  
}
```

- Per API i dettagli, vedere [CreateGroup](#) in AWS CLI Command Reference.

## create-resource

Il seguente esempio di codice mostra come utilizzare `create-resource`.

### AWS CLI

Per creare una nuova risorsa

Il `create-resource` comando seguente crea una nuova risorsa (sala riunioni) per l'organizzazione specificata.

```
aws workmail create-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --name exampleRoom1 \  
  --resource-type MeetingRoom
```

```
--type ROOM
```

Output:

```
{  
  "ResourceId": "r-7afe0efbade843a58cdc10251fce992c"  
}
```

- Per API i dettagli, vedere [CreateResource](#) in AWS CLI Command Reference.

## create-user

Il seguente esempio di codice mostra come utilizzare `create-user`.

### AWS CLI

Per creare un nuovo utente

Il `create-user` comando seguente crea un nuovo utente.

```
aws workmail create-user \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --name exampleName \  
  --display-name exampleDisplayName \  
  --password examplePa$$w0rd
```

Output:

```
{  
  "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333"  
}
```

- Per API i dettagli, vedere [CreateUser](#) in AWS CLI Command Reference.

## delete-access-control-rule

Il seguente esempio di codice mostra come utilizzare `delete-access-control-rule`.

### AWS CLI

Per eliminare una regola di controllo dell'accesso

L'`delete-access-control-rule` seguente elimina la regola di controllo degli accessi specificata dall' WorkMail organizzazione Amazon specificata.

```
aws workmail delete-access-control-rule \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
  --name "myRuLe"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Working with Access Control Rules](#) nella Amazon WorkMail Administrator Guide.

- Per API i dettagli, consulta [DeleteAccessControlRule AWS CLI](#) Command Reference.

## **delete-alias**

Il seguente esempio di codice mostra come utilizzare `delete-alias`.

### AWS CLI

Per eliminare un alias

Il `delete-alias` comando seguente elimina l'alias per l'entità specificata (utente o gruppo).

```
aws workmail delete-alias \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \  
  --alias exampleAlias@site.awsapps.com
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DeleteAlias](#) in AWS CLI Command Reference.

## **delete-group**

Il seguente esempio di codice mostra come utilizzare `delete-group`.

### AWS CLI

Per eliminare un gruppo esistente

Il `delete-group` comando seguente elimina un gruppo esistente da Amazon WorkMail.

```
aws workmail delete-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-333334444-4444
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [DeleteGroup AWS CLI Command Reference](#).

## delete-mailbox-permissions

Il seguente esempio di codice mostra come utilizzare `delete-mailbox-permissions`.

### AWS CLI

Per eliminare le autorizzazioni delle cassette postali

Il `delete-mailbox-permissions` comando seguente elimina le autorizzazioni delle cassette postali precedentemente concesse a un utente o a un gruppo. L'entità rappresenta l'utente proprietario della cassetta postale e l'assegnatario rappresenta l'utente o il gruppo per il quale eliminare le autorizzazioni.

```
aws workmail delete-mailbox-permissions \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1122222222-2222233333-333334444-4444 \  
  --grantee-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Questo comando non produce alcun output.

- Per i API dettagli, vedere [DeleteMailboxPermissions](#) in Command Reference.AWS CLI

## delete-resource

Il seguente esempio di codice mostra come utilizzare `delete-resource`.

### AWS CLI

Per eliminare una risorsa esistente

Il `delete-resource` comando seguente elimina una risorsa esistente da Amazon WorkMail.

```
aws workmail delete-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id S-1-1-11-1122222222-2222233333-333334444-4444
```

```
--organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
--resource-id r-7afe0efbade843a58cdc10251fce992c
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [DeleteResource AWS CLI Command Reference](#).

## **delete-user**

Il seguente esempio di codice mostra come utilizzare `delete-user`.

### AWS CLI

Per eliminare un utente

Il `delete-user` comando seguente elimina l'utente specificato da Amazon WorkMail e da tutti i sistemi successivi.

```
aws workmail delete-user \  
--organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
--user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [DeleteUser AWS CLI Command Reference](#).

## **deregister-from-work-mail**

Il seguente esempio di codice mostra come utilizzare `deregister-from-work-mail`.

### AWS CLI

Per disattivare un'entità esistente

Il `deregister-from-work-mail` comando seguente impedisce a un'entità esistente (utente, gruppo o risorsa) di utilizzare Amazon WorkMail.

```
aws workmail deregister-from-work-mail \  
--organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
--entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [DeregisterFromWorkMail AWS CLI Command Reference](#).

## describe-group

Il seguente esempio di codice mostra come utilizzare `describe-group`.

AWS CLI

Per recuperare informazioni per un gruppo

Il `describe-group` comando seguente recupera le informazioni sul gruppo specificato.

```
aws workmail describe-group \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

Output:

```
{  
  "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444",  
  "Name": "exampleGroup1",  
  "State": "ENABLED"  
}
```

- Per API i dettagli, vedere [DescribeGroup](#) in AWS CLI Command Reference.

## describe-organization

Il seguente esempio di codice mostra come utilizzare `describe-organization`.

AWS CLI

Per recuperare informazioni per un'organizzazione

Il `describe-organization` comando seguente recupera le informazioni per l' WorkMail organizzazione Amazon specificata.

```
aws workmail describe-organization \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```



```
--organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Output:

```
{
  "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"
}
```

Per ulteriori informazioni, consulta [Working with Organizations](#) nella Amazon WorkMail Administrator Guide.

- Per API i dettagli, consulta [DescribeOrganization AWS CLI Command Reference](#).

## describe-resource

Il seguente esempio di codice mostra come utilizzare `describe-resource`.

AWS CLI

Per recuperare informazioni su una risorsa

Il `describe-resource` comando seguente recupera informazioni sulla risorsa specificata.

```
aws workmail describe-resource \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
  --resource-id r-7afe0efbade843a58cdc10251fce992c
```

Output:

```
{
  "ResourceId": "r-7afe0efbade843a58cdc10251fce992c",
  "Name": "exampleRoom1",
  "Type": "ROOM",
}
```

```

    "BookingOptions": {
      "AutoAcceptRequests": true,
      "AutoDeclineRecurringRequests": false,
      "AutoDeclineConflictingRequests": true
    },
    "State": "ENABLED"
  }

```

- Per API i dettagli, vedere [DescribeResource](#) in AWS CLI Command Reference.

## describe-user

Il seguente esempio di codice mostra come utilizzare `describe-user`.

### AWS CLI

Per recuperare le informazioni sull'utente

Il `describe-user` comando seguente recupera le informazioni sull'utente specificato.

```

aws workmail describe-user \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333

```

Output:

```

{
  "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333",
  "Name": "exampleUser1",
  "Email": "exampleUser1@site.awsapps.com",
  "DisplayName": "",
  "State": "ENABLED",
  "UserRole": "USER",
  "EnabledDate": 1532459261.827
}

```

- Per API i dettagli, vedere [DescribeUser](#) in AWS CLI Command Reference.

## disassociate-delegate-from-resource

Il seguente esempio di codice mostra come utilizzare `disassociate-delegate-from-resource`.

## AWS CLI

Per rimuovere un membro da una risorsa

Il `disassociate-delegate-from-resource` comando seguente rimuove il membro specificato da una risorsa.

```
aws workmail disassociate-delegate-from-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DisassociateDelegateFromResource](#) in AWS CLI Command Reference.

## `disassociate-member-from-group`

Il seguente esempio di codice mostra come utilizzare `disassociate-member-from-group`.

## AWS CLI

Per rimuovere un membro da un gruppo

Il `disassociate-member-from-group` comando seguente rimuove il membro specificato da un gruppo.

```
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
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [DisassociateMemberFromGroup](#) in AWS CLI Command Reference.

## `get-access-control-effect`

Il seguente esempio di codice mostra come utilizzare `get-access-control-effect`.

## AWS CLI

Per ottenere l'effetto delle regole di controllo degli accessi

L'`get-access-control-effect` seguente recupera l'effetto delle regole di controllo degli accessi WorkMail dell'organizzazione Amazon specificata per l'indirizzo IP, l'azione del protocollo di accesso e l'ID utente specificati.

```
aws workmail get-access-control-effect \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
  --ip-address "192.0.2.0" \  
  --action "WindowsOutlook" \  
  --user-id "S-1-1-11-1111111111-222222222-3333333333-3333"
```

Output:

```
{  
  "Effect": "DENY",  
  "MatchedRules": [  
    "myRule"  
  ]  
}
```

Per ulteriori informazioni, consulta [Working with Access Control Rules](#) nella Amazon WorkMail Administrator Guide.

- Per API i dettagli, consulta [GetAccessControlEffect AWS CLI Command Reference](#).

## `get-mailbox-details`

Il seguente esempio di codice mostra come utilizzare `get-mailbox-details`.

## AWS CLI

Per ottenere i dettagli della cassetta postale di un utente

Il `get-mailbox-details` comando seguente recupera i dettagli sulla cassetta postale dell'utente specificato.

```
aws workmail get-mailbox-details \  

```

```
--organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
--user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Output:

```
{  
  "MailboxQuota": 51200,  
  "MailboxSize": 0.03890800476074219  
}
```

Per ulteriori informazioni, consulta [Managing User Accounts](#) nella Amazon WorkMail Administrator Guide.

- Per API i dettagli, consulta [GetMailboxDetails AWS CLI Command Reference](#).

## list-access-control-rules

Il seguente esempio di codice mostra come utilizzare `list-access-control-rules`.

AWS CLI

Per elencare le regole di controllo degli accessi

L'`list-access-control-rules` seguente elenca le regole di controllo degli accessi per l' WorkMail organizzazione Amazon specificata.

```
aws workmail list-access-control-rules \  
--organization-id m-n1pq2345678r901st2u3vx45x6789yza
```

Output:

```
{  
  "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
    }
]
}
```

Per ulteriori informazioni, consulta [Working with Access Control Rules](#) nella Amazon WorkMail Administrator Guide.

- Per API i dettagli, consulta [ListAccessControlRules AWS CLI Command Reference](#).

## list-aliases

Il seguente esempio di codice mostra come utilizzare `list-aliases`.

### AWS CLI

Per elencare gli alias di un membro

Il `list-aliases` comando seguente elenca gli alias per il membro specificato (utente o gruppo).

```
aws workmail list-aliases \
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Output:

```
{
  "Aliases": [
    "exampleAlias@site.awsapps.com",
    "exampleAlias1@site.awsapps.com"
  ]
}
```

- Per API i dettagli, vedere [ListAliases](#) in AWS CLI Command Reference.

## list-group-members

Il seguente esempio di codice mostra come utilizzare `list-group-members`.

### AWS CLI

Per elencare i membri del gruppo

Il `list-group-members` comando seguente elenca i membri del gruppo specificato.

```
aws workmail list-group-members \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

Output:

```
{  
  "Members": [  
    {  
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",  
      "Name": "exampleUser1",  
      "Type": "USER",  
      "State": "ENABLED",  
      "EnabledDate": 1532459261.827  
    }  
  ]  
}
```

- Per API i dettagli, vedere [ListGroupMembers](#) in AWS CLI Command Reference.

## list-groups

Il seguente esempio di codice mostra come utilizzare `list-groups`.

### AWS CLI

Per recuperare un elenco di gruppi

Il `list-groups` comando seguente recupera i riepiloghi dei gruppi nell'organizzazione specificata.

```
aws workmail list-groups \  

```

```
--organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Output:

```
{
  "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"
    }
  ]
}
```

- Per API i dettagli, vedere [ListGroups](#) in AWS CLI Command Reference.

## list-mailbox-permissions

Il seguente esempio di codice mostra come utilizzare `list-mailbox-permissions`.

AWS CLI

Per recuperare le autorizzazioni delle cassette postali

Il `list-mailbox-permissions` comando seguente recupera le autorizzazioni della cassetta postale associate alla cassetta postale dell'entità specificata.

```
aws workmail list-mailbox-permissions \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

Output:

```
{
  "Permissions": [
    {
```



```
        "GranteeId": "S-1-1-11-1122222222-2222233333-3333334444-4444",
        "GranteeType": "USER",
        "PermissionValues": [
            "FULL_ACCESS"
        ]
    }
]
```

- Per i API dettagli, vedere [ListMailboxPermissions](#) in Command Reference.AWS CLI

## list-organizations

Il seguente esempio di codice mostra come utilizzare `list-organizations`.

### AWS CLI

Per recuperare un elenco di organizzazioni

Il `list-organizations` comando seguente recupera i riepiloghi delle organizzazioni non eliminate.

```
aws workmail list-organizations
```

Output:

```
{
  "OrganizationSummaries": [
    {
      "OrganizationId": "m-d281d0a2fd824be5b6cd3d3ce909fd27",
      "Alias": "exampleAlias",
      "State": "Active"
    }
  ]
}
```

- Per API i dettagli, vedere [ListOrganizations](#) in AWS CLI Command Reference.

## list-resource-delegates

Il seguente esempio di codice mostra come utilizzare `list-resource-delegates`.

## AWS CLI

Per elencare i delegati di una risorsa

Il `list-resource-delegates` comando seguente recupera i delegati associati alla risorsa specificata.

```
aws workmail list-resource-delegates \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-68bf2d3b1c0244aab7264c24b9217443
```

Output:

```
{  
  "Delegates": [  
    {  
      "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",  
      "Type": "USER"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [ListResourceDelegates](#) in AWS CLI Command Reference.

## list-resources

Il seguente esempio di codice mostra come utilizzare `list-resources`.

### AWS CLI

Per recuperare un elenco di risorse

Il `list-resources` comando seguente recupera i riepiloghi delle risorse per l'organizzazione specificata.

```
aws workmail list-resources \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Output:

```
{
```

```
"Resources": [  
  {  
    "Id": "r-7afe0efbade843a58cdc10251fce992c",  
    "Name": "exampleRoom1",  
    "Type": "ROOM",  
    "State": "ENABLED"  
  }  
]
```

- Per API i dettagli, vedere [ListResources](#) in AWS CLI Command Reference.

## list-tags-for-resource

Il seguente esempio di codice mostra come utilizzare `list-tags-for-resource`.

### AWS CLI

Per elencare i tag di una risorsa

L'`list-tags-for-resource` esempio seguente elenca i tag per l' WorkMail organizzazione Amazon specificata.

```
aws workmail list-tags-for-resource \  
  --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-  
n1pq2345678r901st2u3vx45x6789yza
```

Output:

```
{  
  "Tags": [  
    {  
      "Key": "priority",  
      "Value": "1"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Tagging an Organization](#) nella Amazon WorkMail Administrator Guide.

- Per API i dettagli, consulta [ListTagsForResource AWS CLI Command Reference](#).

## list-users

Il seguente esempio di codice mostra come utilizzare `list-users`.

### AWS CLI

Per recuperare un elenco di utenti

Il `list-users` comando seguente recupera i riepiloghi degli utenti dell'organizzazione specificata.

```
aws workmail list-users \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

Output:

```
{  
  "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"  
    }  
  ]  
}
```

- Per API i dettagli, vedere [ListUsers](#) in AWS CLI Command Reference.

## put-access-control-rule

Il seguente esempio di codice mostra come utilizzare `put-access-control-rule`.

## AWS CLI

Per inserire una nuova regola di controllo degli accessi

L'`put-access-control-rule` seguente nega all'utente specificato l'accesso all'WorkMail organizzazione Amazon specificata.

```
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
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Working with Access Control Rules](#) nella Amazon WorkMail Administrator Guide.

- Per API i dettagli, consulta [PutAccessControlRule AWS CLI Command Reference](#).

## `put-mailbox-permissions`

Il seguente esempio di codice mostra come utilizzare `put-mailbox-permissions`.

## AWS CLI

Per impostare le autorizzazioni delle cassette postali

Il `put-mailbox-permissions` comando seguente imposta le autorizzazioni di accesso complete per l'assegnatario specificato (utente o gruppo). L'entità rappresenta il proprietario della cassetta postale.

```
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
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [PutMailboxPermissions](#) in AWS CLI Command Reference.

## register-to-work-mail

Il seguente esempio di codice mostra come utilizzare `register-to-work-mail`.

### AWS CLI

Per registrare un'entità esistente o disattivata

Il `register-to-work-mail` comando seguente consente all'entità esistente specificata (utente, gruppo o risorsa) di utilizzare Amazon 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
```

Questo comando non produce alcun output.

- Per API i dettagli, consulta [RegisterToWorkMail AWS CLI Command Reference](#).

## reset-password

Il seguente esempio di codice mostra come utilizzare `reset-password`.

### AWS CLI

Per reimpostare la password di un utente

Il `reset-password` comando seguente reimposta la password per l'utente specificato.

```
aws workmail reset-password \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \  
  --password examplePa$$w0rd
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [ResetPassword](#) in AWS CLI Command Reference.

## tag-resource

Il seguente esempio di codice mostra come utilizzare `tag-resource`.

## AWS CLI

Per applicare un tag a una risorsa

L'itag-resourceesempio seguente applica un tag con chiave «priority» e valore «1» all' WorkMail organizzazione Amazon specificata.

```
aws workmail tag-resource \  
  --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-  
n1pq2345678r901st2u3vx45x6789yza \  
  --tags "Key=priority, Value=1"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging an Organization](#) nella Amazon WorkMail Administrator Guide.

- Per API i dettagli, consulta [TagResource AWS CLI](#) Command Reference.

## untag-resource

Il seguente esempio di codice mostra come utilizzare untag-resource.

## AWS CLI

Per rimuovere i tag da una risorsa

L'untag-resourceesempio seguente rimuove il tag specificato dall' WorkMail organizzazione Amazon specificata.

```
aws workmail untag-resource \  
  --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-  
n1pq2345678r901st2u3vx45x6789yza \  
  --tag-keys "priority"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tagging an Organization](#) nella Amazon WorkMail Administrator Guide.

- Per API i dettagli, consulta [UntagResource AWS CLI](#) Command Reference.

## update-mailbox-quota

Il seguente esempio di codice mostra come utilizzare `update-mailbox-quota`.

### AWS CLI

Per aggiornare la quota della cassetta postale di un utente

Il `update-mailbox-quota` comando seguente modifica la quota della cassetta postale dell'utente specificato.

```
aws workmail update-mailbox-quota \  
  --organization-id m-n1pq2345678r901st2u3vx45x6789yza \  
  --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \  
  --mailbox-quota 40000
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Managing User Accounts](#) nella Amazon WorkMail Administrator Guide.

- Per API i dettagli, consulta [UpdateMailboxQuota AWS CLI Command Reference](#).

## update-primary-email-address

Il seguente esempio di codice mostra come utilizzare `update-primary-email-address`.

### AWS CLI

Per aggiornare un indirizzo e-mail principale

Il `update-primary-email-address` comando seguente aggiorna l'indirizzo e-mail principale dell'entità specificata (utente, gruppo o risorsa).

```
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
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [UpdatePrimaryEmailAddress](#) in AWS CLI Command Reference.



## update-resource

Il seguente esempio di codice mostra come utilizzare `update-resource`.

### AWS CLI

Per aggiornare una risorsa

Il `update-resource` comando seguente aggiorna il nome della risorsa specificata.

```
aws workmail update-resource \  
  --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \  
  --resource-id r-7afe0efbade843a58cdc10251fce992c \  
  --name exampleRoom2
```

Questo comando non produce alcun output.

- Per API i dettagli, vedere [UpdateResource](#) in AWS CLI Command Reference.

## Esempi di Amazon WorkMail Message Flow con AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando Amazon WorkMail Message Flow. AWS Command Line Interface

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, puoi vedere le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Azioni](#)

### Azioni

#### **get-raw-message-content**

Il seguente esempio di codice mostra come utilizzare `get-raw-message-content`.

## AWS CLI

Per ottenere il contenuto non elaborato di un messaggio di posta elettronica

L'get-raw-message-content seguente ottiene il contenuto non elaborato di un messaggio di posta elettronica in transito e lo invia a un file di testo denominato test.

```
aws workmailmessageflow get-raw-message-content \  
  --message-id a1b2cd34-ef5g-6h7j-k18m-npq9012345rs \  
  test
```

Contenuto del file test dopo l'esecuzione del comando:

```
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+--
```

Per ulteriori informazioni, consulta [Recupero del contenuto dei messaggi con AWS Lambda nella Amazon Administrator Guide](#). WorkMail

- Per API i dettagli, consulta Command [GetRawMessageContent](#)Reference AWS CLI .

## WorkSpaces esempi utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando AWS Command Line Interface with WorkSpaces.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### **create-tags**

Il seguente esempio di codice mostra come utilizzare `create-tags`.

AWS CLI

Per aggiungere tag a un Workspace

L'`create-tags` seguente aggiunge i tag specificati a quelli specificati Workspace.

```
aws workspaces create-tags \  
  --resource-id ws-dk1xzzr417 \  
  --tags Key=Department,Value=Finance
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tag WorkSpaces resources](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [CreateTags AWS CLI Command Reference](#).

## `create-workspaces`

Il seguente esempio di codice mostra come utilizzare `create-workspaces`.

### AWS CLI

Esempio 1: Per creare un AlwaysOn Workspace

L'`create-workspaces` seguente crea un file AlwaysOn Workspace per l'utente specificato, utilizzando la directory e il pacchetto specificati.

```
aws workspaces create-workspaces \  
  --workspaces DirectoryId=d-926722edaf,UserName=Mateo,BundleId=wsb-0zsvgp8fc
```

Output:

```
{  
  "FailedRequests": [],  
  "PendingRequests": [  
    {  
      "WorkspaceId": "ws-kcqms853t",  
      "DirectoryId": "d-926722edaf",  
      "UserName": "Mateo",  
      "State": "PENDING",  
      "BundleId": "wsb-0zsvgp8fc"  
    }  
  ]  
}
```

## Esempio 2: creare un AutoStop WorkSpace

L'create-workspacesesempio seguente crea un file AutoStop WorkSpace per l'utente specificato, utilizzando la directory e il pacchetto specificati.

```
aws workspaces create-workspaces \
  --
workspaces DirectoryId=d-926722edaf,UserName=Mary,BundleId=wsb-0zsvgp8fc,WorkspaceProperties
```

Output:

```
{
  "FailedRequests": [],
  "PendingRequests": [
    {
      "WorkspaceId": "ws-dk1x zr417",
      "DirectoryId": "d-926722edaf",
      "UserName": "Mary",
      "State": "PENDING",
      "BundleId": "wsb-0zsvgp8fc"
    }
  ]
}
```

## Esempio 3: creare un file disaccoppiato dall'utente WorkSpace

L'create-workspacesesempio seguente crea un utente disaccoppiato WorkSpace impostando il nome utente su e specificando un nome[UNDEFINED], un ID di directory e un ID di WorkSpace pacchetto.

```
aws workspaces create-workspaces \
  --workspaces
DirectoryId=d-926722edaf,UserName=''[UNDEFINED]'' ,WorkspaceName=MaryWorkspace1,BundleId=wsb
```

Output:

```
{
  "FailedRequests": [],
  "PendingRequests": [
    {
      "WorkspaceId": "ws-abcd1234",
```

```
        "DirectoryId": "d-926722edaf",
        "UserName": "[UNDEFINED]",
        "State": "PENDING",
        "BundleId": "wsb-0zsvgp8fc",
        "WorkspaceName": "MaryWorkspace1"
    }
]
}
```

Per ulteriori informazioni, consulta [Launch a virtual desktop](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [CreateWorkspaces AWS CLI](#) Command Reference.

## delete-tags

Il seguente esempio di codice mostra come utilizzare `delete-tags`.

### AWS CLI

Per eliminare un tag da un Workspace

L'esempio seguente elimina il tag specificato dal tag specificato Workspace.

```
aws workspaces delete-tags \
  --resource-id ws-dk1xzt417 \
  --tag-keys Department
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Tag WorkSpaces resources](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [DeleteTags AWS CLI](#) Command Reference.

## deregister-workspace-directory

Il seguente esempio di codice mostra come utilizzare `deregister-workspace-directory`.

### AWS CLI

Per annullare la registrazione di una directory

L'`deregister-workspace-directory` seguente annulla la registrazione della directory specificata.

```
aws workspaces deregister-workspace-directory \  
  --directory-id d-926722edaf
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Registrazione una directory con WorkSpaces](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [DeregisterWorkspaceDirectory AWS CLI Command Reference](#).

## describe-tags

Il seguente esempio di codice mostra come utilizzare `describe-tags`.

### AWS CLI

Per descrivere i tag per un Workspace

L'`describe-tags` seguente descrive i tag per il valore specificato Workspace.

```
aws workspaces describe-tags \  
  --resource-id ws-dk1xzz417
```

Output:

```
{  
  "TagList": [  
    {  
      "Key": "Department",  
      "Value": "Finance"  
    }  
  ]  
}
```

Per ulteriori informazioni, consulta [Tag WorkSpaces resources](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [DescribeTags AWS CLI Command Reference](#).

## describe-workspace-bundles

Il seguente esempio di codice mostra come utilizzare `describe-workspace-bundles`.

### AWS CLI

Per elencare i pacchetti forniti da Amazon

L'esempio seguente elenca i nomi e IDs i pacchetti forniti da Amazon, in formato tabella e ordinati per nome.

```
aws workspaces describe-workspace-bundles \
  --owner AMAZON \
  --query "Bundles[*].[Name, BundleId]"
```

Output:

```
[
  [
    "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"
  ],
  ...
]
```

Per ulteriori informazioni, consulta i [WorkSpaces pacchetti e le immagini](#) nella Amazon WorkSpaces Administration Guide.



- Per API i dettagli, consulta [DescribeWorkspaceBundles AWS CLI Command Reference](#).

## describe-workspace-directories

Il seguente esempio di codice mostra come utilizzare `describe-workspace-directories`.

### AWS CLI

Per descrivere una directory registrata

L'esempio seguente descrive la directory registrata specificata.

```
aws workspaces describe-workspace-directories \  
  --directory-ids d-926722edaf
```

Output:

```
{  
  "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"
    }
}
]
}

```

Per ulteriori informazioni, consulta [Manage directories WorkSpaces](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [DescribeWorkspaceDirectories AWS CLI Command Reference](#).

## describe-workspaces-connection-status

Il seguente esempio di codice mostra come utilizzare `describe-workspaces-connection-status`.

### AWS CLI

Per descrivere lo stato della connessione di un WorkSpace

L'esempio seguente descrive lo stato della connessione dell'oggetto specificato WorkSpace.

```
aws workspaces describe-workspaces-connection-status \
```

```
--workspace-ids ws-dk1xzr417
```

Output:

```
{
  "WorkspacesConnectionStatus": [
    {
      "WorkspaceId": "ws-dk1xzr417",
      "ConnectionState": "CONNECTED",
      "ConnectionStateCheckTimestamp": 1662526214.744
    }
  ]
}
```

Per ulteriori informazioni, consulta [Administer your WorkSpaces](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [DescribeWorkspacesConnectionStatus AWS CLI Command Reference](#).

## describe-workspaces

Il seguente esempio di codice mostra come utilizzare `describe-workspaces`.

AWS CLI

Descrivere un WorkSpace

L'`describe-workspaces` esempio seguente descrive quanto specificato WorkSpace.

```
aws workspaces describe-workspaces \
  --workspace-ids ws-dk1xzr417
```

Output:

```
{
  "Workspaces": [
    {
      "WorkspaceId": "ws-dk1xzr417",
      "DirectoryId": "d-926722edaf",
      "UserName": "Mary",
      "IpAddress": "172.16.0.175",

```

```

    "State": "STOPPED",
    "BundleId": "wsb-0zsvgp8fc",
    "SubnetId": "subnet-500d5819",
    "ComputerName": "WSAMZN-RBSLTDD9",
    "WorkspaceProperties": {
      "RunningMode": "AUTO_STOP",
      "RunningModeAutoStopTimeoutInMinutes": 60,
      "RootVolumeSizeGib": 80,
      "UserVolumeSizeGib": 10,
      "ComputeTypeName": "VALUE"
    },
    "ModificationStates": []
  }
]
}

```

Per ulteriori informazioni, consulta [Administer your WorkSpaces](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [DescribeWorkspaces AWS CLI](#) Command Reference.

## migrate-workspace

Il seguente esempio di codice mostra come utilizzare `migrate-workspace`.

### AWS CLI

Per migrare un WorkSpace

L'esempio seguente migra il pacchetto specificato nel WorkSpace pacchetto specificato.

```

aws workspaces migrate-workspace \
  --source-workspace-id ws-dk1x zr417 \
  --bundle-id wsb-j4d ky1gs4

```

Output:

```

{
  "SourceWorkspaceId": "ws-dk1x zr417",
  "TargetWorkspaceId": "ws-x5h1l bkp5"
}

```

Per ulteriori informazioni, consulta [Migrate a WorkSpace](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [MigrateWorkspace AWS CLI](#) Command Reference.

## **modify-workspace-creation-properties**

Il seguente esempio di codice mostra come utilizzare `modify-workspace-creation-properties`.

### AWS CLI

Per modificare una proprietà di WorkSpace creazione di una directory

L'`modify-workspace-creation-properties` esempio seguente abilita la `EnableInternetAccess` proprietà per la directory specificata. Ciò consente l'assegnazione automatica degli indirizzi IP pubblici per quelli WorkSpaces creati per la directory.

```
aws workspaces modify-workspace-creation-properties \  
  --resource-id d-926722edaf \  
  --workspace-creation-properties EnableInternetAccess=true
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Update directory details for your WorkSpaces](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [ModifyWorkspaceCreationProperties AWS CLI](#) Command Reference.

## **modify-workspace-properties**

Il seguente esempio di codice mostra come utilizzare `modify-workspace-properties`.

### AWS CLI

Per modificare la modalità di esecuzione di un WorkSpace

L'`modify-workspace-properties` esempio seguente imposta la modalità di esecuzione del file specificato WorkSpace su `AUTO_STOP`.

```
aws workspaces modify-workspace-properties \  
  --workspace-id ws-dk1xzzr417 \  
  --execution-mode AUTO_STOP
```

```
--workspace-properties RunningMode=AUTO_STOP
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Modify a Workspace](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [ModifyWorkspaceProperties AWS CLI Command Reference](#).

## modify-workspace-state

Il seguente esempio di codice mostra come utilizzare `modify-workspace-state`.

### AWS CLI

Per modificare lo stato di un Workspace

L'esempio seguente imposta lo stato del Workspace su `ADMIN_MAINTENANCE`.

```
aws workspaces modify-workspace-state \  
  --workspace-id ws-dk1xzi417 \  
  --workspace-state ADMIN_MAINTENANCE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta la sezione [Workspace manutenzione](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [ModifyWorkspaceState AWS CLI Command Reference](#).

## reboot-workspaces

Il seguente esempio di codice mostra come utilizzare `reboot-workspaces`.

### AWS CLI

Per riavviare un Workspace

L'esempio seguente riavvia il file specificato. Workspace

```
aws workspaces reboot-workspaces \  
  --workspace-id ws-dk1xzi417 \  
  --workspace-name Workspace
```

```
--reboot-workspace-requests ws-dk1xzzr417
```

Output:

```
{  
  "FailedRequests": []  
}
```

Per ulteriori informazioni, consulta [Reboot a WorkSpace](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [RebootWorkspaces AWS CLI](#) Command Reference.

## rebuild-workspaces

Il seguente esempio di codice mostra come utilizzare `rebuild-workspaces`.

AWS CLI

Per ricostruire un WorkSpace

L'`rebuild-workspaces` esempio seguente ricostruisce quanto specificato. WorkSpace

```
aws workspaces rebuild-workspaces \  
  --rebuild-workspace-requests ws-dk1xzzr417
```

Output:

```
{  
  "FailedRequests": []  
}
```

Per ulteriori informazioni, consulta [Rebuild a WorkSpace](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [RebuildWorkspaces AWS CLI](#) Command Reference.

## register-workspace-directory

Il seguente esempio di codice mostra come utilizzare `register-workspace-directory`.

## AWS CLI

Per registrare una directory

L'`register-workspace-directory` seguente registra la directory specificata per l'utilizzo con Amazon WorkSpaces.

```
aws workspaces register-workspace-directory \  
  --directory-id d-926722edaf \  
  --no-enable-work-docs
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Registrare una directory con WorkSpaces](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [RegisterWorkspaceDirectory AWS CLI Command Reference](#).

## `restore-workspace`

Il seguente esempio di codice mostra come utilizzare `restore-workspace`.

### AWS CLI

Per ripristinare un Workspace

L'`restore-workspace` seguente ripristina il valore specificato Workspace.

```
aws workspaces restore-workspace \  
  --workspace-id ws-dk1xzi417
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Restore a Workspace](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [RestoreWorkspace AWS CLI Command Reference](#).

## `start-workspaces`

Il seguente esempio di codice mostra come utilizzare `start-workspaces`.



## AWS CLI

Per avviare un AutoStop WorkSpace

L'`start-workspaces` esempio seguente avvia il valore specificato WorkSpace. WorkSpace Deve avere una modalità di esecuzione di AutoStop.

```
aws workspaces start-workspaces \  
  --start-workspace-requests WorkspaceId=ws-dk1xzr417
```

Output:

```
{  
  "FailedRequests": []  
}
```

Per ulteriori informazioni, consulta [Stop and start an AutoStop WorkSpace](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [StartWorkspaces AWS CLI Command Reference](#).

## **stop-workspaces**

Il seguente esempio di codice mostra come utilizzare `stop-workspaces`.

## AWS CLI

Per fermare un AutoStop WorkSpace

L'`stop-workspaces` esempio seguente interrompe il valore specificato WorkSpace. WorkSpace Deve avere una modalità di esecuzione di AutoStop.

```
aws workspaces stop-workspaces \  
  --stop-workspace-requests WorkspaceId=ws-dk1xzr417
```

Output:

```
{  
  "FailedRequests": []  
}
```

Per ulteriori informazioni, consulta [Stop and start an AutoStop Workspace](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [StopWorkspaces AWS CLI](#) Command Reference.

## terminate-workspaces

Il seguente esempio di codice mostra come utilizzare `terminate-workspaces`.

### AWS CLI

Per terminare un Workspace

L'`terminate-workspaces` esempio seguente termina l'area di lavoro specificata.

```
aws workspaces terminate-workspaces \  
  --terminate-workspace-requests ws-dk1xzt417
```

Output:

```
{  
  "FailedRequests": []  
}
```

Per ulteriori informazioni, consulta [Delete a Workspace](#) nella Amazon WorkSpaces Administration Guide.

- Per API i dettagli, consulta [TerminateWorkspaces AWS CLI](#) Command Reference.

## Esempi di raggi X utilizzando AWS CLI

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando il AWS Command Line Interface con X-Ray.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### batch-traces-get

Il seguente esempio di codice mostra come utilizzare `batch-traces-get`.

#### AWS CLI

Per ottenere un elenco di tracce

L'`batch-get-traces` seguente recupera un elenco di tracce specificate da un ID. Il tracciamento completo include un documento per ogni segmento, creato a partire da tutti i documenti di segmento con lo stesso ID di tracciamento ricevuti.

```
aws xray batch-get-traces \
  --trace-ids 1-5d82881a-0a9126e92a73e971eed891b9
```

Output:

```
{
  "Traces": [
    {
      "Id": "1-5d82881a-0a9126e92a73e971eed891b9",
      "Duration": 0.232,
      "Segments": [
        {
          "Id": "54aff5735b12dd28",
          "Document": "{\"id\":\"54aff5735b12dd28\",\"name\":\n\n\"Scorekeep\",\"start_time\":1.568835610432E9,\"end_time\":1.568835610664E9,\n\n\"http\":{\"request\":{\"url\":\"http://scorekeep-env-1.m4fg2pfzpv.us-\n\n-east-2.elasticbeanstalk.com/api/user\",\"method\":\"POST\",\"user_agent\":\n\n\"curl/7.59.0\",\"client_ip\":\"52.95.4.28\",\"x_forwarded_for\":true},\n\n\"response\":{\"status\":200}},\"aws\":{\"elastic_beanstalk\":{\"version_label\n\n\":\"Sample Application-1\",\"deployment_id\":3,\"environment_name\":\"Scorekeep-\n\n-env-1\"},\"ec2\":{\"availability_zone\":\"us-east-2b\",\"instance_id\":\n\n\"i-0e3cf4d2de0f3f37a\"},\"xray\":{\"sdk_version\":\"1.1.0\",\"sdk\":\"X-Ray for\n\n Java\"}},\"service\":{\"runtime\":\"OpenJDK 64-Bit Server VM\",\"runtime_version\n\n\":\"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": []
}

```

Per ulteriori informazioni, vedere [Using the AWS X-Ray API with the AWS CLI nella X-Ray Developer AWS Guide](#).

- Per API i dettagli, vedere [BatchTracesGetin AWS CLI Command Reference](#).

## create-group

Il seguente esempio di codice mostra come utilizzare `create-group`.

### AWS CLI

Per creare un gruppo

L'esempio seguente crea una risorsa di gruppo denominata `AdminGroup`. Il gruppo ottiene un'espressione di filtro che definisce i criteri del gruppo come segmento relativo a un servizio specifico che causa un guasto o un errore.

```
aws xray create-group \
```

```
--group-name "AdminGroup" \  
--filter-expression "service(\"mydomain.com\") {fault OR error}"
```

Output:

```
{  
  "GroupName": "AdminGroup",  
  "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/AdminGroup/123456789",  
  "FilterExpression": "service(\"mydomain.com\") {fault OR error}"  
}
```

Per ulteriori informazioni, vedere [Configuring Sampling, Groups and Encryption Settings with the X-Ray AWS nella X-Ray Developer API Guide](#).AWS

- Per i API dettagli, vedere [CreateGroup](#) in Command Reference.AWS CLI

## create-sampling-rule

Il seguente esempio di codice mostra come utilizzare `create-sampling-rule`.

AWS CLI

Per creare una regola di campionamento

L'esempio seguente crea una regola per controllare il comportamento di campionamento per le applicazioni strumentate. Le regole sono fornite da un JSON file. La maggior parte dei campi delle regole di campionamento sono necessari per creare la regola.

```
aws xray create-sampling-rule \  
--cli-input-json file://9000-base-scorekeep.json
```

Contenuto di `9000-base-scorekeep.json`.

```
{  
  "SamplingRule": {  
    "RuleName": "base-scorekeep",  
    "ResourceARN": "*",  
    "Priority": 9000,  
    "FixedRate": 0.1,  
    "ReservoirSize": 5,  
  }  
}
```

```

    "ServiceName": "Scorekeep",
    "ServiceType": "*",
    "Host": "*",
    "HTTPMethod": "*",
    "URLPath": "*",
    "Version": 1
  }
}

```

Output:

```

{
  "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
  }
}

```

Per ulteriori informazioni, vedere [Configuring Sampling, Groups and Encryption Settings with the X-Ray AWS nella X-Ray Developer API Guide.AWS](#)

- Per i API dettagli, vedere [CreateSamplingRule](#) in Command Reference.AWS CLI

## delete-group

Il seguente esempio di codice mostra come utilizzare `delete-group`.

## AWS CLI

Per eliminare un gruppo

L'`delete-group` esempio seguente elimina la risorsa di gruppo specificata.

```
aws xray delete-group \  
  --group-name "AdminGroup" \  
  --group-arn "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789"
```

Questo comando non produce alcun output.

Per ulteriori informazioni, vedere [Configuring Sampling, Groups and Encryption Settings with the X-Ray AWS nella X-Ray Developer API Guide.AWS](#)

- Per i API dettagli, vedere [DeleteGroup](#) in Command Reference.AWS CLI

## `delete-sampling-rule`

Il seguente esempio di codice mostra come utilizzare `delete-sampling-rule`.

### AWS CLI

Per eliminare una regola di campionamento

L'`delete-sampling-rule` esempio seguente elimina la regola di campionamento specificata. È possibile specificare il gruppo utilizzando il nome o il gruppo. ARN

```
aws xray delete-sampling-rule \  
  --rule-name polling-scorekeep
```

Output:

```
{  
  "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
}
}
```

Per ulteriori informazioni, vedere [Configuring Sampling, Groups and Encryption Settings with the X-Ray AWS nella X-Ray Developer API Guide](#).AWS

- Per i API dettagli, vedere [DeleteSamplingRule](#) in Command Reference.AWS CLI

## get-encryption-config

Il seguente esempio di codice mostra come utilizzare `get-encryption-config`.

### AWS CLI

Per recuperare la configurazione di crittografia

L'`get-encryption-config`esempio seguente recupera la configurazione di crittografia corrente per i dati AWS X-Ray.

```
aws xray get-encryption-config
```

Output:

```
{
  "EncryptionConfig": {
    "KeyId": "ae4aa6d49-a4d8-9df9-a475-4ff6d7898456",
    "Status": "ACTIVE",
    "Type": "NONE"
  }
}
```



Per ulteriori informazioni, vedere [Configuring Sampling, Groups and Encryption Settings with the X-Ray AWS nella X-Ray Developer API Guide](#).AWS

- Per i API dettagli, vedere [GetEncryptionConfigin Command Reference](#).AWS CLI

## get-group

Il seguente esempio di codice mostra come utilizzare `get-group`.

### AWS CLI

Per recuperare un gruppo

L'`get-group`esempio seguente visualizza i dettagli per la risorsa di gruppo specificata. I dettagli includono il nome del gruppo, il gruppo ARN e l'espressione di filtro che definisce i criteri per quel gruppo. I gruppi possono essere recuperati anche daARN.

```
aws xray get-group \  
  --group-name "AdminGroup"
```

Output:

```
{  
  "Group": [  
    {  
      "GroupName": "AdminGroup",  
      "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/  
AdminGroup/123456789",  
      "FilterExpression": "service(\"mydomain.com\") {fault OR error}"  
    }  
  ]  
}
```

Per ulteriori informazioni, vedere [Configuring Sampling, Groups and Encryption Settings with the X-Ray AWS nella X-Ray Developer API Guide](#).AWS

- Per i API dettagli, vedere [GetGroupin Command Reference](#).AWS CLI

## get-groups

Il seguente esempio di codice mostra come utilizzare `get-groups`.

## AWS CLI

Per recuperare tutti i gruppi

L'esempio seguente visualizza i dettagli per tutti i gruppi attivi.

```
aws xray get-groups
```

Output:

```
{
  "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"
    }
  ]
}
```

Per ulteriori informazioni, vedere [Configuring Sampling, Groups and Encryption Settings with the X-Ray AWS nella X-Ray Developer API Guide](#).AWS

- Per i API dettagli, vedere [GetGroups](#) in Command Reference.AWS CLI

## get-sampling-rules

Il seguente esempio di codice mostra come utilizzare `get-sampling-rules`.

## AWS CLI

Per recuperare tutte le regole di campionamento

L'`get-sampling-rules`esempio seguente mostra i dettagli di tutte le regole di campionamento disponibili. :

**aws xray get-sampling-rules**

## Output:

```
{
  "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
    }
  ]
}

```

Per ulteriori informazioni, vedere [Using Sampling Rules with the API X-Ray nella X-Ray Developer AWS Guide](#).

- Per API i dettagli, vedere [GetSamplingRules](#) in AWS CLI Command Reference.

## get-sampling-targets

Il seguente esempio di codice mostra come utilizzare `get-sampling-targets`.

### AWS CLI

Per richiedere una quota di campionamento

L'`get-sampling-targets` esempio seguente richiede una quota di campionamento per le regole utilizzate dal servizio per campionare le richieste. La risposta di AWS X-Ray include una quota che può essere utilizzata invece di prendere in prestito dal serbatoio.

```
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 } ]'
```

Output:

```
{
  "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": []
}
```

Per ulteriori informazioni, vedere [Using Sampling Rules with the API X-Ray nella X-Ray Developer AWS Guide](#).

- Per API i dettagli, vedere [GetSamplingTargets](#) in AWS CLI Command Reference.

## get-service-graph

Il seguente esempio di codice mostra come utilizzare `get-service-graph`.

### AWS CLI

Per ottenere un grafico dei servizi

L'esempio seguente visualizza un documento entro un periodo di tempo specificato che descrive i servizi che elaborano le richieste in entrata e i servizi a valle che richiamano di conseguenza. :

```
aws xray get-service-graph \  
  --start-time 1568835392.0 \  
  --end-time 1568835446.0
```

Output:

```
{  
  "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
}
```

Per ulteriori informazioni, vedere [Using the AWS X-Ray API with the AWS CLI nella X-Ray Developer AWS Guide](#).

- Per API i dettagli, vedere [GetServiceGraphin](#) AWS CLI Command Reference.

## get-trace-summaries

Il seguente esempio di codice mostra come utilizzare `get-trace-summaries`.

### AWS CLI

Per ottenere un riepilogo della traccia

L'esempio seguente recupera i IDs metadati per le tracce disponibili entro un periodo di tempo specificato.

```
aws xray get-trace-summaries \  
  --start-time 1568835392.0 \  
  --end-time 1568835446.0
```

Output:

```
[  
  "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"  
]
```

Per ulteriori informazioni, vedere [Using the AWS X-Ray API with the AWS CLI nella X-Ray Developer AWS Guide](#).

- Per API i dettagli, vedere [GetTraceSummaries](#) in AWS CLI Command Reference.

## put-encryption-config

Il seguente esempio di codice mostra come utilizzare `put-encryption-config`.

### AWS CLI

Per aggiornare la configurazione di crittografia



Quanto segue `put-encryption-config` `example` aggiorna la configurazione di crittografia per i dati AWS X-Ray per utilizzare la chiave AWS gestita di default KMS `aws/xray`.

```
aws xray put-encryption-config \
  --type KMS \
  --key-id alias/aws/xray
```

Output:

```
{
  "EncryptionConfig": {
    "KeyId": "arn:aws:kms:us-west-2:123456789012:key/c234g4e8-39e9-4gb0-84e2-
b0ea215cbba5",
    "Status": "UPDATING",
    "Type": "KMS"
  }
}
```

Per ulteriori informazioni, vedere [Configuring Sampling, Groups and Encryption Settings with the X-Ray AWS nella X-Ray Developer API Guide](#).AWS

- Per i API dettagli, vedere [PutEncryptionConfig](#) in Command Reference.AWS CLI

## put-trace-segments

Il seguente esempio di codice mostra come utilizzare `put-trace-segments`.

AWS CLI

Per caricare un segmento

L' `put-trace-segments` esempio seguente carica i documenti dei segmenti su AWS X-Ray. Il documento del segmento viene utilizzato come elenco di documenti di JSON segmento.

```
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}},\n\"inferred\":true,\n\"aws\":{\n\"consistent_read\":false,\n\"table_name\n\n\":\n\"scorekeep-session-xray\",\n\"operation\":\n\"GetItem\",\n\"request_id\":\n\n\"SCAU230M6M8F038UASGC7785ARVV4KQNS05AEMVJF66Q9ASUAAJG\",\n\n\"resource_names\":\n\n[\n\"scorekeep-session-xray\"]},\n\n\"origin\":\n\n\"AWS::DynamoDB::Table\"}"
```

Output:

```
{
  "UnprocessedTraceSegments": []
}
```

Per ulteriori informazioni, vedere [Invio di dati di traccia a AWS X-Ray nella X-Ray Developer Guide AWS](#).

- Per API i dettagli, vedere [PutTraceSegments](#) in AWS CLI Command Reference.

## update-group

Il seguente esempio di codice mostra come utilizzare `update-group`.

AWS CLI

Per aggiornare un gruppo

L'`update-group` esempio seguente aggiorna i criteri in base ai quali accettare le tracce nel gruppo denominato `AdminGroup`. È possibile specificare il gruppo desiderato utilizzando il nome o il gruppo del gruppo 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}"
```

Output:

```
{
  "GroupName": "AdminGroup",
  "GroupARN": "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789",
  "FilterExpression": "service(\"mydomain.com\") {fault}"
}
```

Per ulteriori informazioni, vedere [Configuring Sampling, Groups and Encryption Settings with the X-Ray AWS nella X-Ray Developer API Guide](#).AWS

- Per i API dettagli, vedere [UpdateGroup](#)in Command Reference.AWS CLI

## update-sampling-rule

Il seguente esempio di codice mostra come utilizzare `update-sampling-rule`.

### AWS CLI

Per aggiornare una regola di campionamento

L'`update-sampling-rule`esempio seguente modifica la configurazione di una regola di campionamento. Le regole vengono utilizzate da un JSON file. Sono obbligatori solo i campi in fase di aggiornamento.

```
aws xray update-sampling-rule \  
  --cli-input-json file://1000-default.json
```

Contenuto di `1000-default.json`.

```
{  
  "SamplingRuleUpdate": {  
    "RuleName": "Default",  
    "FixedRate": 0.01,  
    "ReservoirSize": 0  
  }  
}
```

Output:

```
{  
  "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  
  }  
]  
}
```

Per ulteriori informazioni, vedere [Configuring Sampling, Groups and Encryption Settings with the X-Ray AWS nella X-Ray Developer API Guide.AWS](#)

- Per i API dettagli, vedere [UpdateSamplingRule](#) in Command Reference.AWS CLI

## AWS CLI con esempi di codice di script Bash

Gli esempi di codice in questo argomento mostrano come usare lo script AWS Command Line Interface with Bash con. AWS

Gli elementi di base sono esempi di codice che mostrano come eseguire le operazioni essenziali all'interno di un servizio.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto nei relativi scenari.

Gli scenari sono esempi di codice che mostrano come eseguire attività specifiche richiamando più funzioni all'interno di un servizio o combinandole con altre Servizi AWS.

### Servizi

- [Esempi AWS CLI di DynamoDB utilizzati con lo script Bash](#)
- [EC2Esempi di Amazon che utilizzano AWS CLI lo script Bash](#)
- [HealthImaging esempi di utilizzo AWS CLI con lo script Bash](#)

- [IAM esempi di utilizzo AWS CLI con lo script Bash](#)
- [Esempi di Amazon S3 che utilizzano lo script AWS CLI Bash](#)
- [AWS STS esempi di utilizzo AWS CLI con lo script Bash](#)

## Esempi AWS CLI di DynamoDB utilizzati con lo script Bash

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando lo script AWS Command Line Interface with Bash con DynamoDB.

Le basi sono esempi di codice che mostrano come eseguire le operazioni essenziali all'interno di un servizio.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto nei relativi scenari.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Nozioni di base](#)
- [Azioni](#)


## Nozioni di base

Impara le nozioni di base

L'esempio di codice seguente mostra come:

- Crea una tabella in grado di contenere i dati del filmato.
- Inserisci, ottieni e aggiorna un singolo filmato nella tabella.
- Scrivi i dati del filmato sulla tabella a partire da un JSON file di esempio.
- Esegui una query sui filmati che sono stati rilasciati in un dato anno.
- Cerca i filmati che sono stati distribuiti in diversi anni.
- Elimina un filmato dalla tabella, quindi elimina la tabella.

## AWS CLI con lo script Bash

 Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Lo scenario introduttivo di DynamoDB.

```
#####
# 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"
  fi
fi

```

```

        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
}

```

Le funzioni DynamoDB utilizzate in questo scenario.

```

#####
# 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
}

```

Le funzioni di utility utilizzate in questo scenario.

```

#####
# 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
}
```

- Per API i dettagli, consulta i seguenti argomenti in AWS CLI Command Reference.
  - [BatchWriteItem](#)
  - [CreateTable](#)
  - [DeleteItem](#)
  - [DeleteTable](#)
  - [DescribeTable](#)
  - [GetItem](#)
  - [PutItem](#)
  - [Query](#)
  - [Scan](#)
  - [UpdateItem](#)

## Azioni

### BatchGetItem

Il seguente esempio di codice mostra come utilizzare `BatchGetItem`.

AWS CLI con lo script Bash

#### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
```

```

# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}
```

- Per API i dettagli, vedere [BatchGetItem](#) in AWS CLI Command Reference.

## BatchWriteItem

Il seguente esempio di codice mostra come utilizzare `BatchWriteItem`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
```

```

# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}

```

- Per API i dettagli, vedere [BatchWriteItem](#) in AWS CLI Command Reference.



## CreateTable

Il seguente esempio di codice mostra come utilizzare `CreateTable`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}

```

- Per API i dettagli, vedere [CreateTable](#) in AWS CLI Command Reference.

## DeleteItem

Il seguente esempio di codice mostra come utilizzare `DeleteItem`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}

```


- Per API i dettagli, vedere [DeleteItem](#) in AWS CLI Command Reference.

## DeleteTable

Il seguente esempio di codice mostra come utilizzare DeleteTable.



## AWS CLI con lo script Bash

 Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}

```

- Per API i dettagli, vedere [DeleteTable](#) in AWS CLI Command Reference.

## DescribeTable

Il seguente esempio di codice mostra come utilizzare `DescribeTable`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}

```

- Per API i dettagli, vedere [DescribeTable](#) in AWS CLI Command Reference.

## GetItem

Il seguente esempio di codice mostra come utilizzare `GetItem`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}
```

- Per API i dettagli, vedere [GetItem](#) in AWS CLI Command Reference.

## ListTables

Il seguente esempio di codice mostra come utilizzare `ListTables`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}
```

Le funzioni di utilità utilizzate in questo esempio.

```
#####
# 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
}

```

- Per API i dettagli, vedere [ListTables](#) in AWS CLI Command Reference.

## PutItem

Il seguente esempio di codice mostra come utilizzare PutItem.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}

```


- Per API i dettagli, vedere [PutItem](#) in AWS CLI Command Reference.

## Query

Il seguente esempio di codice mostra come utilizzare Query.



## AWS CLI con lo script Bash

 Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}

```

- Per API i dettagli, vedi [Query](#) in AWS CLI Command Reference.

## Scan

Il seguente esempio di codice mostra come utilizzare `Scan`.

### AWS CLI con lo script Bash

#### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}
```

Le funzioni di utilità utilizzate in questo esempio.

```
#####
# 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
}
```

- Per API i dettagli, consulta [Scan](#) in AWS CLI Command Reference.

## UpdateItem

Il seguente esempio di codice mostra come utilizzare `UpdateItem`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}
```

- Per API i dettagli, vedere [UpdateItem](#) in AWS CLI Command Reference.

## EC2 Esempi di Amazon che utilizzano AWS CLI lo script Bash

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando lo script AWS Command Line Interface with Bash con AmazonEC2.

Le basi sono esempi di codice che mostrano come eseguire le operazioni essenziali all'interno di un servizio.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto nei relativi scenari.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Nozioni di base](#)
- [Azioni](#)

### Nozioni di base

Impara le nozioni di base

L'esempio di codice seguente mostra come:

- Creare una coppia di chiavi e un gruppo di sicurezza.
- Seleziona un'Amazon Machine Image (AMI) e un tipo di istanza compatibile, quindi crea un'istanza.
- Arrestare e riavviare l'istanza.

- Associazione di un indirizzo IP elastico all'istanza
- Connettiti alla tua istanza conSSH, quindi pulisci le risorse.

## AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Esegui uno scenario interattivo al prompt dei comandi.

```
#####
# 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

```

```
    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

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

    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
}

```

Le funzioni DynamoDB utilizzate in questo scenario.

```

#####
# 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
        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

    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)."
```

```
    echo ""
}

# Retrieve the calling parameters.
while getopts "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
}
```

```
#####
# 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 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
}

#####
# 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
    fi
}

```

```

    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
```

```

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'"
    fi
}
#####
```

```

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
}

```

```

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
}

```

```

    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
                ;;
            \?)
                echo "Invalid parameter"
                usage
                return 1
        esac
    done
}

```



```

        ;;
    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
    fi
}
```

```

    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
        esac
    done
}

```

```

        ;;
    \?)
        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
}

```

Le funzioni di utility utilizzate in questo scenario.

```

#####
# 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
}
```

- Per API i dettagli, consulta i seguenti argomenti in AWS CLI Command Reference.
  - [AllocateAddress](#)
  - [AssociateAddress](#)
  - [AuthorizeSecurityGroupIngress](#)
  - [CreateKeyPair](#)
  - [CreateSecurityGroup](#)
  - [DeleteKeyPair](#)
  - [DeleteSecurityGroup](#)
  - [DescribeImages](#)
  - [DescribeInstanceTypes](#)

- [DescribeInstances](#)
- [DescribeKeyPairs](#)
- [DescribeSecurityGroups](#)
- [DisassociateAddress](#)
- [ReleaseAddress](#)
- [RunInstances](#)
- [StartInstances](#)
- [StopInstances](#)
- [TerminateInstances](#)
- [UnmonitorInstances](#)

## Azioni

### AllocateAddress

Il seguente esempio di codice mostra come utilizzare `AllocateAddress`.

AWS CLI con lo script Bash

#### Note

C'è altro da fare. [GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.](#)

```
#####  
# 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'."
    fi
}

```



```

    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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}
```

- Per API i dettagli, vedere [AllocateAddress](#) in AWS CLI Command Reference.

## AssociateAddress

Il seguente esempio di codice mostra come utilizzare `AssociateAddress`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}
```

- Per API i dettagli, vedere [AssociateAddress](#) in AWS CLI Command Reference.

## AuthorizeSecurityGroupIngress

Il seguente esempio di codice mostra come utilizzare `AuthorizeSecurityGroupIngress`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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."
    fi
}

```



```

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
}

```

- Per API i dettagli, vedere [AuthorizeSecurityGroupIngress](#) in AWS CLI Command Reference.

## CreateKeyPair

Il seguente esempio di codice mostra come utilizzare `CreateKeyPair`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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
        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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}
```

- Per API i dettagli, vedere [CreateKeyPair](#) in AWS CLI Command Reference.

## CreateSecurityGroup

Il seguente esempio di codice mostra come utilizzare `CreateSecurityGroup`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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

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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}


```

- Per API i dettagli, vedere [CreateSecurityGroup](#) in AWS CLI Command Reference.

## DeleteKeyPair

Il seguente esempio di codice mostra come utilizzare `DeleteKeyPair`.

## AWS CLI con lo script Bash

 Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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"
        esac
    done
}
```



```

        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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}

```

- Per API i dettagli, vedere [DeleteKeyPair](#) in AWS CLI Command Reference.

## DeleteSecurityGroup

Il seguente esempio di codice mostra come utilizzare `DeleteSecurityGroup`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
```

```

# 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
}

```

```

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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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."
    fi
}

```

```
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
}
```

- Per API i dettagli, vedere [DeleteSecurityGroup](#) in AWS CLI Command Reference.

## DescribeImages

Il seguente esempio di codice mostra come utilizzare `DescribeImages`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}

```

- Per API i dettagli, vedere [DescribeImages](#) in AWS CLI Command Reference.

## DescribeInstanceTypes

Il seguente esempio di codice mostra come utilizzare `DescribeInstanceTypes`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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
fi
```

```
    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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}

```

- Per API i dettagli, vedere [DescribeInstanceTypes](#) in AWS CLI Command Reference.

## DescribeInstances

Il seguente esempio di codice mostra come utilizzare `DescribeInstances`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}
```

- Per API i dettagli, vedere [DescribeInstances](#) in AWS CLI Command Reference.

## DescribeKeyPairs

Il seguente esempio di codice mostra come utilizzare `DescribeKeyPairs`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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"
                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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}

```

- Per API i dettagli, vedere [DescribeKeyPairs](#) in AWS CLI Command Reference.

## DescribeSecurityGroups

Il seguente esempio di codice mostra come utilizzare `DescribeSecurityGroups`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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 "$response"

return 0
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}

```

- Per API i dettagli, vedere [DescribeSecurityGroups](#) in AWS CLI Command Reference.

## DisassociateAddress

Il seguente esempio di codice mostra come utilizzare `DisassociateAddress`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. [GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.](#)

```

#####
# 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
}
```

Le funzioni di utilità utilizzate in questo esempio.

```
#####
# 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
}

```

- Per API i dettagli, vedere [DisassociateAddress](#) in AWS CLI Command Reference.

## ReleaseAddress

Il seguente esempio di codice mostra come utilizzare `ReleaseAddress`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```
#####
```

```

# 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
}

```

- Per API i dettagli, vedere [ReleaseAddress](#) in AWS CLI Command Reference.

## RunInstances

Il seguente esempio di codice mostra come utilizzare `RunInstances`.

### AWS CLI con lo script Bash

#### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}
```

- Per API i dettagli, vedere [RunInstances](#) in AWS CLI Command Reference.

## StartInstances

Il seguente esempio di codice mostra come utilizzare `StartInstances`.

## AWS CLI con lo script Bash

 Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
            )
        esac
    done
}
```

```

        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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}


```

- Per API i dettagli, vedere [StartInstances](#) in AWS CLI Command Reference.

## StopInstances

Il seguente esempio di codice mostra come utilizzare `StopInstances`.

## AWS CLI con lo script Bash

 Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
            )
        esac
    done
}
```

```

        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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}
```

- Per API i dettagli, vedere [StopInstances](#) in AWS CLI Command Reference.

## TerminateInstances

Il seguente esempio di codice mostra come utilizzare `TerminateInstances`.

## AWS CLI con lo script Bash

 Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

Le funzioni di utilità utilizzate in questo esempio.

```

#####
# 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
}

```

- Per API i dettagli, vedere [TerminateInstances](#) in AWS CLI Command Reference.

## HealthImaging esempi di utilizzo AWS CLI con lo script Bash

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando lo script AWS Command Line Interface with Bash con. HealthImaging

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

## Argomenti

- [Azioni](#)

## Azioni

### CreateDatastore

Il seguente esempio di codice mostra come utilizzare `CreateDatastore`.

#### AWS CLI con lo script Bash

```
#####
# 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
}

```

- Per API i dettagli, vedere [CreateDatastore](#) in AWS CLI Command Reference.

### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

## DeleteDatastore

Il seguente esempio di codice mostra come usare `DeleteDatastore`.

AWS CLI con lo script Bash

```

#####
# 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
}

```

- Per API i dettagli, vedere [DeleteDatastore](#) in AWS CLI Command Reference.

### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

## GetDatastore

Il seguente esempio di codice mostra come usare `GetDatastore`.

AWS CLI con lo script Bash

```

#####
# 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
}

```

- Per API i dettagli, vedere [GetDatastore](#) in AWS CLI Command Reference.

### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

## ListDatastores

Il seguente esempio di codice mostra come usare `ListDatastores`.

AWS CLI con lo script Bash

```

#####
# 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
}
```

- Per API i dettagli, vedere [ListDatastores](#) in AWS CLI Command Reference.

#### Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

## IAM esempi di utilizzo AWS CLI con lo script Bash

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando lo script AWS Command Line Interface with Bash con IAM.

Le basi sono esempi di codice che mostrano come eseguire le operazioni essenziali all'interno di un servizio.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto nei relativi scenari.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Nozioni di base](#)
- [Azioni](#)



## Nozioni di base

### Impara le nozioni di base

Il seguente esempio di codice mostra come creare un utente e assumere un ruolo.

#### Warning

Per evitare rischi per la sicurezza, non utilizzate IAM gli utenti per l'autenticazione quando sviluppate software appositamente creato o lavorate con dati reali. Utilizza invece la federazione con un provider di identità come [AWS IAM Identity Center](#).

- Crea un utente che non disponga di autorizzazioni.
- Crea un ruolo che conceda l'autorizzazione per elencare i bucket Amazon S3 per l'account.
- Aggiungi una policy per consentire all'utente di assumere il ruolo.
- Assumi il ruolo ed elenca i bucket S3 utilizzando le credenziali temporanee, quindi ripulisci le risorse.

### AWS CLI con lo script Bash

#### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####  
# 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"
    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
}

```

Le IAM funzioni utilizzate in questo scenario.

```

#####
# 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
}

#####
# 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
        esac
    done
}

```

```
        ;;
    \?)
        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
        return 1
    fi
}
```

```

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
```

```

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
}
```

- Per API i dettagli, vedere i seguenti argomenti in *AWS CLI Command Reference*.
  - [AttachRolePolicy](#)
  - [CreateAccessKey](#)
  - [CreatePolicy](#)
  - [CreateRole](#)
  - [CreateUser](#)
  - [DeleteAccessKey](#)
  - [DeletePolicy](#)
  - [DeleteRole](#)
  - [DeleteUser](#)
  - [DeleteUserPolicy](#)
  - [DetachRolePolicy](#)
  - [PutUserPolicy](#)

## Azioni

### **AttachRolePolicy**

Il seguente esempio di codice mostra come utilizzare `AttachRolePolicy`.

AWS CLI con lo script Bash

#### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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 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
}
```

- Per API i dettagli, vedere [AttachRolePolicy](#) in AWS CLI Command Reference.

## CreateAccessKey

Il seguente esempio di codice mostra come utilizzare `CreateAccessKey`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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 getopts "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
}

```

- Per API i dettagli, vedere [CreateAccessKey](#) in AWS CLI Command Reference.

## CreatePolicy

Il seguente esempio di codice mostra come utilizzare `CreatePolicy`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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
    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"
}

```

- Per API i dettagli, vedere [CreatePolicy](#) in AWS CLI Command Reference.

## CreateRole

Il seguente esempio di codice mostra come utilizzare `CreateRole`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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 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
}
```

- Per API i dettagli, vedere [CreateRole](#) in AWS CLI Command Reference.

## CreateUser

Il seguente esempio di codice mostra come utilizzare CreateUser.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
    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
}

```

- Per API i dettagli, vedere [CreateUser](#) in AWS CLI Command Reference.

## DeleteAccessKey

Il seguente esempio di codice mostra come utilizzare `DeleteAccessKey`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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 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
}
```

- Per API i dettagli, vedere [DeleteAccessKey](#) in AWS CLI Command Reference.

## DeletePolicy

Il seguente esempio di codice mostra come utilizzare DeletePolicy.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

- Per API i dettagli, vedere [DeletePolicy](#) in AWS CLI Command Reference.

## DeleteRole

Il seguente esempio di codice mostra come utilizzare `DeleteRole`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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 getopt "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
}
```

- Per API i dettagli, vedere [DeleteRole](#) in AWS CLI Command Reference.

## DeleteUser

Il seguente esempio di codice mostra come utilizzare `DeleteUser`.

## AWS CLI con lo script Bash

 Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

- Per API i dettagli, vedere [DeleteUser](#) in AWS CLI Command Reference.

## DetachRolePolicy

Il seguente esempio di codice mostra come utilizzare `DetachRolePolicy`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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
}
```

- Per API i dettagli, vedere [DetachRolePolicy](#) in AWS CLI Command Reference.

## GetUser

Il seguente esempio di codice mostra come utilizzare `GetUser`.

## AWS CLI con lo script Bash

 Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

- Per API i dettagli, vedere [GetUser](#) in AWS CLI Command Reference.

## ListAccessKeys

Il seguente esempio di codice mostra come utilizzare `ListAccessKeys`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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."
        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
}

```

- Per API i dettagli, vedere [ListAccessKeys](#) in AWS CLI Command Reference.

## ListUsers

Il seguente esempio di codice mostra come utilizzare `ListUsers`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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
}
```

- Per API i dettagli, vedere [ListUsers](#) in AWS CLI Command Reference.

## Esempi di Amazon S3 che utilizzano lo script AWS CLI Bash

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando lo script AWS Command Line Interface with Bash con Amazon S3.

Le basi sono esempi di codice che mostrano come eseguire le operazioni essenziali all'interno di un servizio.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Mentre le azioni mostrano come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel loro contesto nei relativi scenari.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

### Argomenti

- [Nozioni di base](#)
- [Azioni](#)

## Nozioni di base

Impara le nozioni di base

L'esempio di codice seguente mostra come:

- Crea un bucket e carica un file in tale bucket.
- Scaricare un oggetto da un bucket.
- Copiare un oggetto in una sottocartella in un bucket.
- Elencare gli oggetti in un bucket.
- Elimina il bucket e tutti gli oggetti in esso contenuti.

## AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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

  if delete_bucket "$bucket_name"; then
    echo "Deleted the bucket $bucket_name"
  else
    result=1
  fi
fi
```

```

    return $result
}

```

Le funzioni di Amazon S3 utilizzate in questo scenario.

```

#####
# 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}" ;;
        esac
    done
}

```

```
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
}
```


- Per API i dettagli, consulta i seguenti argomenti in AWS CLI Command Reference.
  - [CopyObject](#)
  - [CreateBucket](#)
  - [DeleteBucket](#)
  - [DeleteObjects](#)
  - [GetObject](#)
  - [ListObjectsV2](#)
  - [PutObject](#)

## Azioni

### CopyObject

Il seguente esempio di codice mostra come usare `CopyObject`.

## AWS CLI con lo script Bash

 Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}
```

- Per API i dettagli, vedere [CopyObject](#) in AWS CLI Command Reference.

## CreateBucket

Il seguente esempio di codice mostra come utilizzare `CreateBucket`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}
```

- Per API i dettagli, vedere [CreateBucket](#) in AWS CLI Command Reference.

## DeleteBucket

Il seguente esempio di codice mostra come utilizzare DeleteBucket.

### AWS CLI con lo script Bash

#### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

- Per API i dettagli, vedere [DeleteBucket](#) in AWS CLI Command Reference.

## DeleteObject

Il seguente esempio di codice mostra come utilizzare `DeleteObject`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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
}
```

- Per API i dettagli, vedere [DeleteObject](#) in AWS CLI Command Reference.

## DeleteObjects

Il seguente esempio di codice mostra come utilizzare DeleteObjects.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}


```

- Per API i dettagli, vedere [DeleteObjects](#) in AWS CLI Command Reference.

## GetObject

Il seguente esempio di codice mostra come utilizzare `GetObject`.

## AWS CLI con lo script Bash

 Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}
```

- Per API i dettagli, vedere [GetObject](#) in AWS CLI Command Reference.

## HeadBucket

Il seguente esempio di codice mostra come utilizzare HeadBucket.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
# 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
}

```

- Per API i dettagli, vedere [HeadBucket](#) in AWS CLI Command Reference.

## ListObjectsV2

Il seguente esempio di codice mostra come utilizzare `ListObjectsV2`.

AWS CLI con lo script Bash

### Note

C'è altro da fare. [GitHub](#) Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

#####
# 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
}

```

- Per API i dettagli, vedi [ListObjectsV2](#) in AWS CLI Command Reference.

## PutObject

Il seguente esempio di codice mostra come utilizzare PutObject.

AWS CLI con lo script Bash

### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####
```

```

# 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
}

```

- Per API i dettagli, vedere [PutObject](#) in AWS CLI Command Reference.

## AWS STS esempi di utilizzo AWS CLI con lo script Bash

I seguenti esempi di codice mostrano come eseguire azioni e implementare scenari comuni utilizzando lo script AWS Command Line Interface with Bash con. AWS STS

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni di servizio, è possibile visualizzare le azioni nel contesto negli scenari correlati.

Ogni esempio include un collegamento al codice sorgente completo, in cui è possibile trovare istruzioni su come configurare ed eseguire il codice nel contesto.

Argomenti

- [Azioni](#)

Azioni

### AssumeRole

Il seguente esempio di codice mostra come utilizzareAssumeRole.

AWS CLI con lo script Bash

#### Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
#####  
# 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} ;;
            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
}
```

- Per API i dettagli, vedere [AssumeRole](#) in AWS CLI Command Reference.

# Sicurezza nel AWS CLI

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS e te. Il [modello di responsabilità condivisa](#) descrive questo approccio come sicurezza del cloud e sicurezza nel cloud:

- **Sicurezza del cloud:** AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei [AWS Programmi di AWS conformità dei Programmi di conformità](#) dei di . Per ulteriori informazioni sui programmi di conformità applicabili AWS Command Line Interface, consulta [AWS Servizi nell'ambito del programma di conformitàAWS](#) .
- **Sicurezza nel cloud:** la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa quando si utilizza AWS Command Line Interface (AWS CLI). I seguenti argomenti mostrano come configurarlo per AWS CLI soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche come utilizzare il AWS CLI per monitorare e proteggere AWS le tue risorse.

## Argomenti

- [Protezione dei dati nel AWS CLI](#)
- [Identity and Access Management](#)
- [Convalida della conformità per questo AWS prodotto o servizio](#)
- [Resilienza per questo AWS prodotto o servizio](#)
- [Sicurezza dell'infrastruttura per questo AWS prodotto o servizio](#)
- [Applicazione di una versione minima di TLS for AWS CLI](#)

# Protezione dei dati nel AWS CLI

Il modello di [responsabilità AWS condivisa modello](#) di si applica alla protezione dei dati in AWS Command Line Interface. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, consulta la sezione [Privacy dei dati FAQ](#). Per informazioni sulla protezione dei dati in Europa, consulta il [Modello di responsabilitàAWS condivisa e GDPR](#) il post sul blog sulla AWS sicurezza.

Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS le credenziali e di configurare i singoli utenti con AWS IAM Identity Center o AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Usa SSL/TLS per comunicare con AWS le risorse. Richiediamo TLS 1.2 e consigliamo TLS 1.3.
- Configurazione API e registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'uso dei CloudTrail percorsi per registrare AWS le attività, consulta [Lavorare con i CloudTrail percorsi](#) nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di FIPS 140-3 moduli crittografici convalidati per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint. FIPS Per ulteriori informazioni sugli FIPS endpoint disponibili, vedere [Federal Information Processing Standard \(\) 140-3. FIPS](#)

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori AWS CLI o Servizi AWS utilizzi in altro modo la console, API AWS CLI, o. AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Se fornisci un URL a un server esterno, ti consigliamo vivamente di non includere le informazioni sulle credenziali URL per convalidare la tua richiesta a quel server.

## Crittografia dei dati

Una caratteristica fondamentale di qualsiasi servizio sicuro è che le informazioni vengano crittografate quando non sono utilizzate attivamente.

### Crittografia a riposo

Di per sé AWS CLI non memorizza alcun dato del cliente diverso dalle credenziali necessarie per interagire con i AWS servizi per conto dell'utente.

Se utilizzi il AWS CLI per richiamare un AWS servizio che trasmette i dati dei clienti al computer locale per l'archiviazione, consulta il capitolo Sicurezza e conformità della Guida per l'utente del servizio per informazioni su come tali dati vengono archiviati, protetti e crittografati.

### Crittografia in transito

Per impostazione predefinita, tutti i dati trasmessi dal computer client che esegue gli endpoint AWS CLI e del AWS servizio vengono crittografati inviandoli tramite una HTTPS connessione/. TLS

Non è necessario fare nulla per abilitare l'uso diHTTPS/TLS. È sempre abilitato a meno che non si disattiva esplicitamente per un singolo comando utilizzando l'opzione della riga di comando `--no-verify-ssl`.

## Identity and Access Management

AWS Identity and Access Management (IAM) è un dispositivo Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. IAM gli amministratori controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse. AWS IAM è un dispositivo Servizio AWS che puoi utilizzare senza costi aggiuntivi.

### Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso con policy](#)
- [Come Servizi AWS lavorare con IAM](#)
- [Risoluzione dei problemi relativi AWS all'identità e all'accesso](#)

## Destinatari

Il modo in cui usi AWS Identity and Access Management (IAM) varia a seconda del lavoro che svolgi. AWS

**Utente del servizio:** se lo utilizzi Servizi AWS per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più AWS funzionalità per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità di AWS, consulta [Risoluzione dei problemi relativi AWS all'identità e all'accesso](#) o consulta la guida per l'utente della funzionalità Servizio AWS che stai utilizzando.

**Amministratore del servizio:** se sei responsabile delle AWS risorse della tua azienda, probabilmente hai pieno accesso a AWS. È tuo compito determinare a quali AWS funzionalità e risorse devono accedere gli utenti del servizio. È quindi necessario inviare richieste all'IAM amministratore per modificare le autorizzazioni degli utenti del servizio. Consulta le informazioni contenute in questa pagina per comprendere i concetti di base di IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con AWS, consulta la guida per l'utente del Servizio AWS dispositivo che stai utilizzando.

**IAM amministratore:** se sei un IAM amministratore, potresti voler conoscere i dettagli su come scrivere politiche per gestire l'accesso AWS. Per visualizzare esempi di policy AWS basate sull'identità che puoi utilizzare IAM, consulta la guida per l'utente del programma Servizio AWS che stai utilizzando.

## Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. È necessario autenticarsi (accedere a AWS) come Utente root dell'account AWS, come IAM utente o assumendo un ruolo. IAM

È possibile accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Quando accedi come identità federata, l'amministratore aveva precedentemente configurato la federazione delle identità utilizzando i ruoli. IAM Quando si accede AWS utilizzando la federazione, si assume indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi [Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente](#).

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando () per firmare crittograficamente le tue richieste utilizzando le tue credenziali. CLI Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta [AWS Signature Version 4 per API le richieste](#) nella Guida per l'IAMutente.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta [Autenticazione a più fattori](#) nella Guida per l'AWS IAM Identity Center utente e [Autenticazione a AWS più fattori IAM nella Guida per l'IAMutente](#).

## Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per l'elenco completo delle attività che richiedono l'accesso come utente root, consulta [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'IAMutente.

## Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per informazioni su IAM Identity Center, vedi [Cos'è IAM Identity Center?](#) nella Guida AWS IAM Identity Center per l'utente.

## IAM users and groups

Un [IAMutente](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Laddove possibile, consigliamo di fare affidamento su credenziali temporanee anziché creare IAM utenti con credenziali a lungo termine come password e chiavi di accesso. Tuttavia, se hai casi d'uso specifici che richiedono credenziali a lungo termine con IAM gli utenti, ti consigliamo di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta [Ruotare regolarmente le chiavi di accesso per i casi d'uso che richiedono credenziali a lungo termine](#) nella Guida per l'utente. IAM

Un [IAMgruppo](#) è un'identità che specifica un insieme di utenti. IAM Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile assegnare un nome a un gruppo IAMAdminse concedere a tale gruppo le autorizzazioni per IAM amministrare le risorse.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta [Casi d'uso per IAM gli utenti nella Guida per l'IAMutente](#).

## IAMruoli

Un [IAMruolo](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un IAM utente, ma non è associato a una persona specifica. Per assumere temporaneamente un IAM ruolo in AWS Management Console, puoi [passare da un utente a un IAM ruolo \(console\)](#). È possibile assumere un ruolo chiamando un' AWS APIoperazione AWS CLI or o utilizzando un'operazione personalizzataURL. Per ulteriori informazioni sui metodi di utilizzo dei ruoli, vedere [Metodi per assumere un ruolo](#) nella Guida per l'IAMutente.

IAMi ruoli con credenziali temporanee sono utili nelle seguenti situazioni:

- **Accesso utente federato:** per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per informazioni sui ruoli per la federazione, vedere [Creazione di un ruolo per un provider di identità di terze parti](#) nella Guida per l'IAMutente. Se utilizzi IAM Identity Center, configuri un set di autorizzazioni. Per controllare a cosa possono accedere le identità dopo l'autenticazione, IAM Identity Center correla

il set di autorizzazioni a un ruolo in IAM. Per informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .

- Autorizzazioni IAM utente temporanee: un IAM utente o un ruolo può assumere il IAM ruolo di assumere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso su più account: puoi utilizzare un IAM ruolo per consentire a qualcuno (un responsabile fidato) di un altro account di accedere alle risorse del tuo account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per conoscere la differenza tra ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta la [sezione Accesso alle risorse su più account IAM nella Guida per l'utente IAM](#).
- Accesso tra servizi: alcuni Servizi AWS utilizzano funzionalità in altri. Servizi AWS Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
- Sessioni di accesso inoltrato (FAS): quando utilizzi un IAM utente o un ruolo per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, in combinazione con la richiesta di effettuare richieste Servizio AWS ai servizi downstream. FAS le richieste vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli FAS delle politiche relative alle richieste, consulta [Forward access sessions](#).
- Ruolo di servizio: un ruolo di servizio è un [IAM ruolo](#) che un servizio assume per eseguire azioni per conto dell'utente. Un IAM amministratore può creare, modificare ed eliminare un ruolo di servizio dall'interno IAM. Per ulteriori informazioni, vedere [Creazione di un ruolo per delegare le autorizzazioni a un utente Servizio AWS nella Guida per l'utente IAM](#).
- Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un IAM amministratore può visualizzare, ma non modificare le autorizzazioni per i ruoli collegati al servizio.
- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un IAM ruolo per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che effettuano AWS CLI o richiedono AWS API. Ciò è preferibile alla memorizzazione delle chiavi di accesso all'interno



dell'EC2istanza. Per assegnare un AWS ruolo a un'EC2istanza e renderlo disponibile per tutte le sue applicazioni, create un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull'EC2istanza di ottenere credenziali temporanee. Per ulteriori informazioni, consulta [Usare un IAM ruolo per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella Guida](#) per l'IAMutente.

## Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come JSON documenti. Per ulteriori informazioni sulla struttura e il contenuto dei documenti relativi alle JSON politiche, vedere [Panoramica delle JSON politiche](#) nella Guida per l'IAMutente.

Gli amministratori possono utilizzare AWS JSON le politiche per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un IAM amministratore può creare IAM politiche. L'amministratore può quindi aggiungere le IAM politiche ai ruoli e gli utenti possono assumerli.

IAMle politiche definiscono le autorizzazioni per un'azione indipendentemente dal metodo utilizzato per eseguire l'operazione. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale criterio può ottenere informazioni sul ruolo da AWS Management Console, da o da. AWS CLI AWS API

### Policy basate su identità

I criteri basati sull'identità sono documenti relativi alle politiche di JSON autorizzazione che è possibile allegare a un'identità, ad esempio un IAM utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una politica basata sull'identità, consulta [Definire le IAM autorizzazioni personalizzate con](#) le politiche gestite dal cliente nella Guida per l'utente. IAM

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche

gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli all'interno del tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una politica gestita o una politica in linea, consulta [Scegliere tra politiche gestite e politiche in linea nella Guida](#) per l'IAM utente.

## Policy basate su risorse

Le politiche basate sulle risorse sono documenti di JSON policy allegati a una risorsa. Esempi di politiche basate sulle risorse sono le policy di trust dei IAM ruoli e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite contenute IAM in una policy basata sulle risorse.

## Elenchi di controllo degli accessi (ACLs)

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy. JSON

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la [panoramica di Access control list \(ACL\)](#) nella Amazon Simple Storage Service Developer Guide.

## Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- **Limiti delle autorizzazioni:** un limite di autorizzazioni è una funzionalità avanzata in cui si impostano le autorizzazioni massime che una politica basata sull'identità può concedere a un'entità (utente o ruolo). IAM IAM È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di

queste policy sostituisce l'autorizzazione. [Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità nella Guida per l'utente. IAM IAM](#)

- Politiche di controllo del servizio (SCPs): SCPs sono JSON politiche che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. SCP Limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna Utente root dell'account AWS. Per ulteriori informazioni su Organizations and SCPs, consulta [le politiche di controllo dei servizi](#) nella Guida AWS Organizations per l'utente.
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta [le politiche di sessione](#) nella Guida IAM per l'utente.

## Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per informazioni su come AWS determinare se consentire una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle politiche](#) nella Guida per l'IAM utente.

## Come Servizi AWS lavorare con IAM

Per avere una panoramica generale del Servizi AWS funzionamento della maggior parte delle IAM funzionalità, consulta [AWS i servizi con cui è possibile utilizzare IAM](#) nella Guida per l'IAM utente.

Per informazioni su come utilizzare una larghezza specifica Servizio AWS IAM, consulta la sezione sulla sicurezza della Guida per l'utente del servizio pertinente.

## Risoluzione dei problemi relativi AWS all'identità e all'accesso

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con AWS e IAM.

## Argomenti

- [Non sono autorizzato a eseguire alcuna azione in AWS](#)
- [Non sono autorizzato a eseguire iam: PassRole](#)
- [Voglio consentire a persone esterne a me di accedere Account AWS alle mie AWS risorse](#)

## Non sono autorizzato a eseguire alcuna azione in AWS

Se ricevi un errore che indica che non disponi dell'autorizzazione per eseguire un'operazione, le tue policy devono essere aggiornate in modo che ti sei consentito eseguire tale operazione.

L'errore di esempio seguente si verifica quando l'utente `mateojackson` tenta di utilizzare la console per visualizzare i dettagli su una `my-example-widget` risorsa fittizia ma non dispone delle autorizzazioni fittizie `aws:GetWidget`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
aws:GetWidget on resource: my-example-widget
```

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa `my-example-widget` utilizzando l'azione `aws:GetWidget`.

Se hai bisogno di assistenza, contatta l'amministratore. AWS L'amministratore è la persona che ti ha fornito le credenziali di accesso.

## Non sono autorizzato a eseguire iam: PassRole

Se ricevi un errore che indica che non sei autorizzato a eseguire l'operazione `iam:PassRole`, le tue policy devono essere aggiornate per poter passare un ruolo a AWS.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un IAM utente denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in AWS. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di assistenza, contatta AWS l'amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

## Voglio consentire a persone esterne a me di accedere Account AWS alle mie AWS risorse

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se AWS supporta queste funzionalità, consulta [Come Servizi AWS lavorare con IAM](#)
- Per informazioni su Account AWS come fornire l'accesso alle risorse di tua proprietà, consulta [Fornire l'accesso a un IAM utente di un altro Account AWS utente di tua proprietà](#) nella Guida per l'IAMutente.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta [Fornire l'accesso a persone Account AWS di proprietà di terzi](#) nella Guida per l'IAMutente.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta [Fornire l'accesso agli utenti autenticati esternamente \(federazione delle identità\)](#) nella Guida per l'IAMutente.
- Per conoscere la differenza tra l'utilizzo di ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta la sezione Accesso alle [risorse tra account nella Guida per l'utente](#). IAM IAM

## Convalida della conformità per questo AWS prodotto o servizio

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione [Scope by Compliance Program Servizi AWS](#) e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di [AWS conformità Programmi](#) di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Scaricamento dei report in AWS Artifact](#) .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- [Guide introduttive su sicurezza e conformità](#): queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla AWS sicurezza e la conformità.
- [Architettura per la HIPAA sicurezza e la conformità su Amazon Web Services](#): questo white paper descrive in che modo le aziende possono utilizzare AWS per creare applicazioni idonee. HIPAA

#### Note

Non tutte sono idonee. Servizi AWS HIPAA Per ulteriori informazioni, consulta la [Guida ai servizi HIPAA idonei](#).

- [AWS Risorse per AWS](#) per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe riguardare il tuo settore e la tua località.
- [AWS Guide alla conformità dei clienti](#): comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization ()). ISO
- [Evaluating Resources with Rules](#) nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- [AWS Security Hub](#)— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina [Documentazione di riferimento sui controlli della Centrale di sicurezza](#).
- [Amazon GuardDuty](#): Servizio AWS rileva potenziali minacce ai tuoi carichi di lavoro Account AWS, ai contenitori e ai dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty può aiutarti a soddisfare vari requisiti di conformità, ad esempio PCI DSS soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.
- [AWS Audit Manager](#)— Ciò Servizio AWS consente di verificare continuamente AWS l'utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

Questo AWS prodotto o servizio segue il [modello di responsabilità condivisa](#) attraverso i servizi specifici di Amazon Web Services (AWS) che supporta. Per informazioni sulla sicurezza dei AWS servizi, consulta la [pagina della documentazione sulla sicurezza del AWS servizio](#) e [AWS i servizi che rientrano nell'ambito delle iniziative di AWS conformità previste dal programma di conformità](#).

## Resilienza per questo AWS prodotto o servizio

L'infrastruttura AWS globale è costruita attorno a zone Regioni AWS di disponibilità.

Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti.

Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

[Per ulteriori informazioni su AWS regioni e zone di disponibilità, vedere Global Infrastructure.AWS](#)

Questo AWS prodotto o servizio segue il [modello di responsabilità condivisa](#) attraverso i servizi specifici di Amazon Web Services (AWS) che supporta. Per informazioni sulla sicurezza dei AWS servizi, consulta la [pagina della documentazione sulla sicurezza del AWS servizio](#) e [AWS i servizi che rientrano nell'ambito delle iniziative di AWS conformità previste dal programma di conformità](#).

## Sicurezza dell'infrastruttura per questo AWS prodotto o servizio

Questo AWS prodotto o servizio utilizza servizi gestiti ed è pertanto protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta [AWS Cloud Security](#). Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Si utilizzano API chiamate AWS pubblicate per accedere a questo AWS Prodotto o Servizio attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). Richiediamo TLS 1.2 e consigliamo TLS 1.3.
- Suite di crittografia con perfetta segretezza di inoltro (PFS) come (Ephemeral Diffie-Hellman) o DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.



Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale. IAM O puoi utilizzare [AWS Security Token Service](#) (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Questo AWS prodotto o servizio segue il [modello di responsabilità condivisa](#) attraverso gli specifici servizi Amazon Web Services (AWS) che supporta. Per informazioni sulla sicurezza dei AWS servizi, consulta la [pagina della documentazione sulla sicurezza del AWS servizio](#) e [AWS i servizi che rientrano nell'ambito delle iniziative di AWS conformità previste dal programma di conformità](#).

## Applicazione di una versione minima di TLS for AWS CLI

Quando si utilizza AWS Command Line Interface (AWS CLI), il protocollo Transport Layer Security (TLS) svolge un ruolo cruciale nel proteggere la comunicazione tra e. AWS CLI Servizi AWS Per aumentare la sicurezza durante la comunicazione con AWS i servizi, è necessario utilizzare TLS 1.2 o versioni successive.

La AWS CLI e Servizio AWS può scambiare dati in modo sicuro, con il TLS protocollo che fornisce crittografia, autenticazione e integrità dei dati. Sfruttando il TLS protocollo, AWS CLI garantiscono che le vostre interazioni con Servizi AWS siano protette da accessi non autorizzati e violazioni dei dati, migliorando la sicurezza complessiva del vostro ecosistema. AWS

Il modello di [responsabilità AWS condivisa modello](#) di dei dati in. AWS Command Line Interface Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Servizi AWS. L'utente è responsabile di mantenere il controllo sui contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile delle attività di configurazione e gestione della sicurezza relative a Servizi AWS ciò che utilizza. Per ulteriori informazioni sulla protezione dei dati, vedere [the section called "Protezione dei dati"](#).

Per garantire che la AWS CLI versione 1 non utilizzi alcuna TLS versione precedente alla TLS 1.2, potrebbe essere necessario ricompilare Open per SSL imporre questo valore minimo e quindi ricompilare Python per utilizzare l'Open appena creato. SSL

### Argomenti

- [Determinare i protocolli attualmente supportati](#)
- [Compila Open SSL e Python](#)



## Determinare i protocolli attualmente supportati

Innanzitutto, crea un certificato autofirmato da utilizzare per il server di test e SDK Python utilizzando Open. SSL

```
$ openssl req -subj '/CN=localhost' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out cert.pem -days 365
```

Quindi avvia un server di prova utilizzando Open. SSL

```
$ openssl s_server -key key.pem -cert cert.pem -www
```

In una nuova finestra di terminale, crea un ambiente virtuale e installa SDK for Python.

```
$ python3 -m venv test-env
source test-env/bin/activate
pip install botocore
```

Crea un nuovo script Python denominato `check.py` che utilizzi la libreria SDK sottostante HTTP.

```
$ 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'))
```

Eseguire il nuovo script.

```
$ python check.py
```

In questo modo vengono visualizzati i dettagli sulla connessione effettuata. Cercare "Protocol : " nell'output. Se l'output è "TLSv1.2" o successivo, il SDK valore predefinito è v1.2 o successivo. TLS Se si tratta di una versione precedente, è necessario ricompilare Open SSL e ricompilare Python.

Tuttavia, anche se l'installazione di Python è predefinita alla TLS v1.2 o successiva, è comunque possibile per Python rinegoziare a una versione precedente alla v1.2 se il server non supporta la

TLS v1.2 o successiva. TLS Per verificare che Python non rinegozi automaticamente su versioni precedenti, riavviare il server di test con quanto segue.

```
$ openssl s_server -key key.pem -cert cert.pem -no_tls1_3 -no_tls1_2 -www
```

Se utilizzi una versione precedente di Open, potresti non avere il flag disponibile. SSL -no\_tls\_3 In tal caso, rimuovi il flag perché la versione di Open SSL che stai utilizzando non supporta la TLS versione 1.3. Eseguire lo script Python.

```
$ python check.py
```

Se la tua installazione di Python non rinegozia correttamente per versioni precedenti alla TLS 1.2, dovresti ricevere un errore. 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)'))))
```

Se riesci a stabilire una connessione, devi ricompilare Open e SSL Python per disabilitare la negoziazione dei protocolli precedenti alla v1.2. TLS

## Compila Open SSL e Python

Per assicurarti che SDK or AWS CLI non negozii per nulla precedente alla TLS 1.2, devi ricompilare Open e SSL Python. A tale scopo, copiare il contenuto seguente per creare uno script ed eseguirlo.

```
#!/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
```

Questo compila una versione di Python che ha un SSL Open collegato staticamente che non negozia automaticamente nulla prima della 1.2. TLS Questo installa anche Open SSL nella `/opt/openssl-with-min-tls1_2` directory e installa Python nella directory. `/opt/python-with-min-tls1_2` Dopo aver eseguito questo script, confermare l'installazione della nuova versione di Python.

```
$ /opt/python-with-min-tls1_2/bin/python3 --version
```

Questa operazione dovrebbe stampare quanto segue.

```
$ Python 3.8.1
```

Per confermare che questa nuova versione di Python non negozia una versione precedente alla TLS 1.2, riesegui i passaggi dall'[Determinare i protocolli attualmente supportati](#) utilizzo della versione di Python appena installata (ovvero,). `/opt/python-with-min-tls1_2/bin/python3`

# Risoluzione degli errori per AWS CLI

Questa sezione descrive gli errori più comuni e le procedure di risoluzione dei problemi da seguire per risolvere il problema. Ti suggeriamo di seguire prima la [risoluzione generale della risoluzione dei problemi](#).

## Indice

- [Risoluzione dei problemi generali da provare prima](#)
  - [Controlla la formattazione AWS CLI dei comandi](#)
  - [Verifica che il Regione AWS tuo AWS CLI comando stia usando](#)
  - [Conferma che stai utilizzando una versione recente di AWS CLI](#)
  - [Usa l'--debugopzione](#)
  - [Abilita e rivedi i registri della cronologia dei AWS CLI comandi](#)
  - [Conferma che il tuo AWS CLI sia configurato](#)
- [Errori relativi al comando non trovato](#)
- [Il comando `aws --version ""` restituisce una versione diversa da quella installata](#)
- [Il comando `"aws --version"` restituisce una versione dopo la disinstallazione di AWS CLI](#)
- [Ha AWS CLI elaborato un comando con un nome di parametro incompleto](#)
- [Errori di accesso negato](#)
- [Credenziali non valide ed errori chiave](#)
- [La firma non corrisponde agli errori](#)
- [Nessun errore rilevato dalla console Windows](#)
- [SSLerrori nei certificati](#)
- [Errori non validi JSON](#)
- [Risorse aggiuntive](#)

## Risoluzione dei problemi generali da provare prima

Se ricevi un errore o riscontri un problema con il AWS CLI, ti suggeriamo i seguenti suggerimenti generali per aiutarti a risolvere il problema.

[Torna all'inizio](#)

## Controlla la formattazione AWS CLI dei comandi

Se ricevi un errore che indica che un comando non esiste o che non riconosce un parametro (`Parameter validation failed`) indicato nella documentazione come disponibile, il comando potrebbe essere formattato in modo errato. Ti suggeriamo di controllare quanto segue:

- Verifica la presenza di errori di ortografia e formattazione nel comando.
- Conferma tutte le [virgolette e l'escape appropriato per il tuo terminale](#) è corretto nel tuo comando.
- Genera uno [AWS CLI scheletro](#) per confermare la struttura dei comandi.
- PerJSON, consultate la sezione aggiuntiva sulla [risoluzione dei problemi relativi ai JSON valori](#). Se riscontri problemi con la JSON formattazione dell'elaborazione del terminale, ti suggeriamo di ignorare le regole di quotazione del terminale utilizzando [Blobs per passare JSON i dati direttamente a. AWS CLI](#)

[Torna all'inizio](#)

## Verifica che il Regione AWS tuo AWS CLI comando stia usando

### Note

È necessario specificare un valore Regione AWS quando si utilizza la AWS CLI, in modo esplicito o impostando una regione predefinita. Per un elenco di tutti gli elementi Regioni AWS che è possibile specificare, consulta [AWS Regioni ed endpoint](#) in. Riferimenti generali di Amazon Web Services I Regione AWS designatori utilizzati da AWS CLI sono gli stessi nomi che vedi negli AWS Management Console URLs endpoint del servizio.

Potrebbero verificarsi errori o risultati imprevisti se uno non Servizio AWS è disponibile per i dati specificati Regione AWS o se le risorse si trovano in un altro. Regione AWS In ordine di precedenza, Regione AWS viene impostato nei seguenti modi:

- L'opzione della `--region` riga di comando.
- La variabile di [AWS\\_DEFAULT\\_REGION](#) ambiente.
- L'impostazione [region](#) del profilo.

Conferma di utilizzare le risorse corrette Regione AWS per le tue risorse.

[Torna all'inizio](#)

## Conferma che stai utilizzando una versione recente di AWS CLI

Se ricevi un errore che indica che un comando non esiste o che non riconosce un parametro che la guida di [AWS CLI riferimento alla AWS CLI versione 2 della guida](#) indica disponibile, verifica innanzitutto che il comando sia formattato correttamente. Se la formattazione è corretta, ti consigliamo di eseguire l'aggiornamento alla versione più recente di AWS CLI. Le versioni aggiornate di AWS CLI vengono rilasciate quasi ogni giorno lavorativo. AWS Nuovi servizi, funzionalità e parametri vengono introdotti in queste nuove versioni di AWS CLI. L'unico modo per ottenere l'accesso a questi nuovi servizi, caratteristiche o parametri è eseguire l'aggiornamento a una versione rilasciata dopo che l'elemento è stato introdotto per la prima volta.

La modalità di aggiornamento della versione di AWS CLI dipende da come è stata originariamente installata, come descritto in [Installare il AWS CLI](#).

Se hai utilizzato uno dei programmi di installazione in dotazione, potresti dover rimuovere l'installazione esistente prima di scaricare e installare la versione più recente per il tuo sistema operativo.

[Torna all'inizio](#)

## Usa l'- -debugopzione

Quando AWS CLI riporta un errore che non comprendi immediatamente o produce risultati inaspettati, puoi ottenere maggiori dettagli sull'errore eseguendo nuovamente il comando con l'- -debugopzione. Con questa opzione, vengono AWS CLI visualizzati i dettagli di ogni passaggio necessario per elaborare il comando. I dettagli nell'output possono aiutarti a determinare quando si verifica l'errore e forniscono indizi su dove è iniziato.

È possibile inviare l'output a un file di testo per una revisione successiva o inviarlo AWS Support quando richiesto.

Quando includi l'- -debugopzione, alcuni dettagli includono:

- Ricerca delle credenziali
- Analisi dei parametri specificati
- Costruzione della richiesta inviata ai server AWS
- Il contenuto della richiesta inviata a AWS

- Il contenuto della risposta non elaborata
- L'output formattato

Ecco un esempio di un comando eseguito con e senza l'opzione `--debug`.

```
$ 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 - boto3.core.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 - boto3.core.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 - boto3.core.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 - boto3.core.loaders - DEBUG - Loading JSON  
file: /home/ec2-user/venv/lib/python3.7/site-packages/boto3/core/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 - boto3.core.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 - boto3.core.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 - boto3.core.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 - boto3.core.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
  5f776d91EXAMPLE9b8cb5eb5d6d4a787a33ae41c8cd6eEXAMPLEca69080e1e1f
2019-08-12 12:36:18,344 - MainThread - botoconfig.auth - DEBUG - StringToSign:
  AWS4-HMAC-SHA256
```

```

20190812T193618Z
20190812/us-east-1/iam/aws4_request
ab7e367eEXAMPLE2769f178ea509978cf8bfa054874b3EXAMPLE8d043fab6cc9
2019-08-12 12:36:18,344 - MainThread - boto.core.auth - DEBUG - Signature:
d85a0EXAMPLEb40164f2f539cdc76d4f294fe822EXAMPLE18ad1ddf58a1a3ce7
2019-08-12 12:36:18,344 - MainThread - boto.core.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 boto.core/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 - boto.core.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 - boto.core.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 - boto.core.hooks - DEBUG - Event needs-
retry.iam.ListGroups: calling handler <boto.core.retryhandler.RetryHandler object at
0x7fdf16e9a780>
2019-08-12 12:36:18,665 - MainThread - boto.core.retryhandler - DEBUG - No retry needed.
2019-08-12 12:36:18,665 - MainThread - boto.core.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"
    }
]
}
```

[Torna all'inizio](#)

## Abilita e rivedi i registri della cronologia dei AWS CLI comandi

È possibile abilitare i registri della cronologia dei AWS CLI comandi utilizzando l'impostazione del [cli\\_history](#) file. Dopo aver abilitato questa impostazione, AWS CLI registra la cronologia dei aws comandi.

È possibile elencare la cronologia utilizzando il `aws history list` comando e utilizzare il `command_ids` risultante nel `aws history show` comando per i dettagli. Per ulteriori informazioni, consulta la guida [aws history](#) di riferimento. AWS CLI

Quando includi l'`--debug` opzione, alcuni dettagli includono:

- API chiamate effettuate a botocore
- Codici di stato
- HTTP richieste
- Headers
- Codici di restituzione

È possibile utilizzare queste informazioni per confermare che i dati dei parametri e API le chiamate si comportano nel modo previsto e quindi dedurre in quale fase del processo il comando non funziona.

[Torna all'inizio](#)

## Conferma che il tuo AWS CLI sia configurato

Se `credentials` i file `config` e l'IAM utente o il ruolo non sono configurati correttamente, possono verificarsi diversi errori. Per ulteriori informazioni sulla risoluzione degli errori relativi ai `credentials` file `config` and o IAM all'utente o ai ruoli, consulta [the section called "Errori di accesso negato"](#) e [the section called "Credenziali non valide ed errori chiave"](#).

[Torna all'inizio](#)

## Errori relativi al comando non trovato

Questo errore indica che il sistema operativo non riesce a trovare il AWS CLI comando. L'installazione potrebbe essere incompleta o richiedere un aggiornamento.

Possibile causa: stai cercando di utilizzare una AWS CLI funzionalità più recente della versione installata o hai una formattazione errata

Esempio di testo di errore:

```
$ 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
....
```

Possono verificarsi diversi errori se il comando è formattato in modo errato o se si utilizza una versione precedente precedente al rilascio della funzionalità. Per ulteriori informazioni sulla risoluzione degli errori relativi a questi due problemi, consulta e. [the section called “Controlla la formattazione AWS CLI dei comandi”](#) [the section called “Conferma che stai utilizzando una versione recente di AWS CLI”](#)

[Torna all'inizio](#)

Possibile causa: il terminale deve essere riavviato dopo l'installazione

Testo di errore di esempio:

```
$ aws --version
command not found: aws
```

Se il aws comando non viene trovato dopo la prima installazione o l'aggiornamento di AWS CLI, potrebbe essere necessario riavviare il terminale per far sì che riconosca eventuali PATH aggiornamenti.

[Torna all'inizio](#)

Possibile causa: l'installazione AWS CLI non è stata completata

Testo di errore di esempio:

```
$ aws --version
command not found: aws
```

Se il `aws` comando non può essere trovato dopo la prima installazione o l'aggiornamento di AWS CLI, è possibile che non sia stato installato completamente. Prova a reinstallarlo seguendo i passaggi relativi alla tua piattaforma in [Installare il AWS CLI](#).

[Torna all'inizio](#)

Possibile causa: AWS CLI non dispone di autorizzazioni (Linux)

Se il `aws` comando non viene trovato dopo la prima installazione o l'aggiornamento AWS CLI su Linux, è possibile che non disponga execute delle autorizzazioni per la cartella in cui è installato. Esegui il seguente comando con la PATH tua AWS CLI installazione, per fornire [chmod](#) le autorizzazioni a: AWS CLI

```
$ sudo chmod -R 755 /usr/local/aws-cli/
```

[Torna all'inizio](#)

Possibile causa: il sistema operativo non **PATH** è stato aggiornato durante l'installazione

Testo di errore di esempio:

```
$ aws --version
command not found: aws
```

Potrebbe essere necessario aggiungere l'eseguibile `aws` alla variabile di ambiente `PATH` del sistema operativo in uso. Per aggiungerlo AWS CLI al tuo `PATH`, usa le seguenti istruzioni per il tuo sistema operativo.

Linux and macOS

1. Individua lo script del profilo della shell nella cartella utente. Se non hai la certezza di quale sia la tua shell, esegui `echo $SHELL`.

```
$ ls -a ~
```

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

- Bash – `.bash_profile`, `.profile`, o `.bash_login`
  - Zsh – `.zshrc`
  - Tcsh – `.tcshrc`, `.cshrc`, o `.login`
2. Aggiungi un comando di esportazione allo script del tuo profilo. Il comando seguente aggiunge il contenitore locale alla PATH variabile corrente.

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

3. Ricarica il profilo aggiornato nella sessione corrente.

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

## Windows

1. In un prompt dei comandi di Windows, utilizzate il `where` comando con il `/R path` parametro per trovare la posizione del `aws` file. I risultati restituiscono tutte le cartelle contenenti `aws`.

```
C:\> where /R c:\ aws  
c:\Program Files\Amazon\AWSCLIV2\aws.exe  
...
```

Per impostazione predefinita, la AWS CLI versione 2 si trova in:

```
c:\Program Files\Amazon\AWSCLIV2\aws.exe
```

2. Premere il tasto Windows e immettere **environment variables**.
3. Dall'elenco dei suggerimenti, scegli Modifica variabili di ambiente per il tuo account.
4. Scegli PATH, quindi scegli Modifica.
5. Aggiungi il percorso che hai trovato nel primo passaggio nel campo Valore variabile, ad esempio ***C:\Program Files\Amazon\AWSCLIV2\aws.exe***.
6. Fai doppio clic su OK per applicare le nuove impostazioni.
7. Chiudi tutte le richieste di comando in esecuzione e riapri la finestra del prompt dei comandi.

[Torna all'inizio](#)

## Il comando `aws --version ""` restituisce una versione diversa da quella installata

È possibile che il terminale restituisca una PATH risposta diversa AWS CLI da quella prevista.

Possibile causa: il terminale deve essere riavviato dopo l'installazione

Se il `aws` comando mostra la versione sbagliata, potrebbe essere necessario riavviare il terminale per riconoscere eventuali PATH aggiornamenti. Tutti i terminali aperti devono essere chiusi, non solo il terminale attivo.

[Torna all'inizio](#)

Possibile causa: il sistema deve essere riavviato dopo l'installazione

Se il `aws` comando mostra la versione sbagliata e il riavvio del terminale non ha funzionato, potrebbe essere necessario riavviare il sistema per far sì che riconosca gli aggiornamenti. PATH

[Torna all'inizio](#)

Possibile causa: sono disponibili più versioni di AWS CLI

Se hai aggiornato AWS CLI e utilizzato un metodo di installazione diverso da quello dell'installazione preesistente, è possibile che vengano installate più versioni. Ad esempio, se su Linux o macOS lo hai utilizzato `pip` per l'installazione corrente, ma hai provato ad aggiornare utilizzando il file di `.pkg` installazione, ciò potrebbe causare alcuni conflitti, specialmente con il PATH riferimento alla versione precedente.

Per risolvere il problema, [disinstalla tutte le versioni di AWS CLI ed esegui](#) un'installazione pulita.

Dopo aver disinstallato tutte le versioni, segui le istruzioni appropriate per il tuo sistema operativo per installare la versione desiderata della [AWS CLI versione 1](#) o della [AWS CLI versione 2](#).

### Note

Se ciò accade dopo aver installato la AWS CLI versione 2 con un'installazione preesistente della AWS CLI versione 1, segui le istruzioni di migrazione contenute nelle Istruzioni di [installazione durante la migrazione dalla AWS CLI versione 1](#).



[Torna all'inizio](#)

## Il comando "**aws --version**" restituisce una versione dopo la disinstallazione di AWS CLI

Ciò si verifica spesso quando ne è ancora AWS CLI installato uno da qualche parte sul sistema.

Possibile causa: il terminale deve essere riavviato dopo la disinstallazione

Se il `aws --version` comando funziona ancora, potrebbe essere necessario riavviare il terminale per far sì che riconosca eventuali aggiornamenti del terminale.

[Torna all'inizio](#)

Possibile causa: AWS CLI sul sistema sono presenti più versioni di o non è stato utilizzato lo stesso metodo di disinstallazione utilizzato per l'installazione originale di AWS CLI

AWS CLI Potrebbe non essere disinstallato correttamente se lo hai disinstallato AWS CLI utilizzando un metodo diverso da quello usato per installarlo o se hai installato più versioni. Ad esempio, se l'hai utilizzata `pip` per l'installazione corrente, devi `pip` disinstallarla. Per risolvere il problema, esegui la disinstallazione AWS CLI utilizzando lo stesso metodo utilizzato per installarla.

1. Segui le istruzioni appropriate per il tuo sistema operativo e il metodo di installazione originale per disinstallare la [AWS CLI versione 1](#) e la [AWS CLI versione 2](#).
2. Chiudi tutti i terminali che hai aperto.
3. Apri il tuo terminale preferito, inserisci il seguente comando e conferma che non viene restituita alcuna versione.

```
$ aws --version
command not found: aws
```

Se hai ancora una versione elencata nell'output, probabilmente AWS CLI è stata installata utilizzando un metodo diverso o esistono più versioni. Se non sai quale metodo hai installato AWS CLI, segui le istruzioni per ogni metodo di disinstallazione per la [AWS CLI versione 1](#) e [la AWS CLI versione 2](#) appropriati al tuo sistema operativo fino a quando non viene ricevuta alcuna versione in uscita.

**Note**

Se hai usato un gestore di pacchetti per installare AWS CLI (pip, aptbrew, ecc.), devi usare lo stesso gestore di pacchetti per disinstallarlo. Assicuratevi di seguire le istruzioni fornite dal gestore di pacchetti su come disinstallare tutte le versioni di un pacchetto.

[Torna all'inizio](#)

## Ha AWS CLI elaborato un comando con un nome di parametro incompleto

Possibile causa: è stata utilizzata un'abbreviazione riconosciuta del parametro AWS CLI

Poiché AWS CLI è stato creato utilizzando Python, AWS CLI utilizza la `argparse` libreria Python, incluso l'argomento. [allow\\_abbrev](#) Le abbreviazioni dei parametri vengono riconosciute da ed elaborate AWS CLI .

Il seguente esempio di [create-change-set](#) comando modifica il nome dello CloudFormation stack. Il parametro `--change-set-n` viene riconosciuto come abbreviazione di `--change-set-name` ed AWS CLI elabora il comando.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-n my-change-set
```

Se l'abbreviazione potrebbe essere composta da più comandi, il parametro non verrà riconosciuto come abbreviazione.

Il seguente esempio di [create-change-set](#) comando modifica il nome dello CloudFormation stack. Il parametro non `--change-set-` è riconosciuto come abbreviazione, poiché esistono più parametri di cui potrebbe essere un'abbreviazione, ad esempio `and. --change-set-name --change-set-type` Pertanto AWS CLI non elabora il comando.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set- my-change-set
```

**⚠ Warning**

Non utilizzate intenzionalmente le abbreviazioni dei parametri. Non sono affidabili e non sono retrocompatibili. Se a un comando vengono aggiunti nuovi parametri che confondono le abbreviazioni, i comandi verranno interrotti.

Inoltre, se il parametro è un argomento a valore singolo, può causare un comportamento imprevisto con i comandi. Se vengono passate più istanze di un argomento a valore singolo, verrà eseguita solo l'ultima istanza. Nell'esempio seguente, il parametro `--filters` è un argomento a valore singolo. I parametri `--filters` e `--filter` sono specificati. Il `--filter` parametro è l'abbreviazione di `--filters`. Ciò comporta l'applicazione di due istanze e `--filters` viene applicato solo l'ultimo `--filter` argomento.

```
$ aws ec2 describe-vpc-peering-connections \
  --filters Name=tag:TagName,Values=VpcPeeringConnection \
  --filter Name=status-code,Values=active
```

Confermate di utilizzare parametri validi prima di eseguire un comando per evitare comportamenti imprevisti.

[Torna all'inizio](#)

## Errori di accesso negato

Possibile causa: il file di AWS CLI programma non dispone dell'autorizzazione di «esecuzione»

Su Linux o macOS, assicurati che il `aws` programma disponga delle autorizzazioni di esecuzione per l'utente chiamante. In genere, le autorizzazioni sono impostate su 755.

Per aggiungere l'autorizzazione di esecuzione per il tuo utente, esegui il comando seguente, sostituendo `~/local/bin/aws` con il percorso del programma sul tuo computer.

```
$ chmod +x ~/local/bin/aws
```

[Torna all'inizio](#)

## Possibile causa: la tua IAM identità non è autorizzata a eseguire l'operazione

Esempio di testo di errore:

```
$ aws s3 ls
An error occurred (AccessDenied) when calling the ListBuckets operation: Access
denied.
```

Quando esegui un AWS CLI comando, AWS le operazioni vengono eseguite per tuo conto, utilizzando credenziali che ti associano a un IAM account o a un ruolo. I criteri allegati devono concedere all'utente l'autorizzazione a richiamare API le azioni che corrispondono ai comandi eseguiti con. AWS CLI

La maggior parte dei comandi chiamano una singola operazione con un nome corrispondente al nome del comando. Tuttavia, comandi personalizzati come le `aws s3 sync` chiamate multipleAPIs. È possibile vedere quale APIs comando chiama utilizzando l'--debugopzione.

Se sei sicuro che l'utente o il ruolo disponga delle autorizzazioni appropriate assegnate dalla policy, assicurati che il AWS CLI comando utilizzi le credenziali previste. Consultate la [sezione successiva sulle credenziali](#) per verificare che le credenziali utilizzate AWS CLI siano quelle previste.

Per informazioni sull'assegnazione IAM delle autorizzazioni, vedere [Panoramica sulla gestione degli accessi: autorizzazioni e politiche](#) nella Guida per l'utente. IAM

[Torna all'inizio](#)

## Credenziali non valide ed errori chiave

Testo di errore di esempio:

```
$ 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.
```

## Possibile causa: AWS CLI sta leggendo credenziali errate o da una posizione inaspettata

È AWS CLI possibile che stia leggendo le credenziali da una posizione diversa da quella prevista o che le informazioni sulla coppia di chiavi non siano corrette. Puoi eseguire `aws configure list` per verificare le credenziali utilizzate.

L'esempio seguente mostra come controllare le credenziali utilizzate per il profilo predefinito.

```
$ 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
```

L'esempio seguente mostra come controllare le credenziali di un profilo denominato.

```
$ 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
```

Per confermare i dettagli della tua key pair, controlla i tuoi credentials file config e. Per ulteriori informazioni sui credentials file config and, consulta [the section called “Impostazioni dei file di configurazione e credenziali in AWS CLI”](#). Per ulteriori informazioni sulle credenziali e sull'autenticazione, inclusa la precedenza delle credenziali, vedere. [Credenziali di autenticazione e accesso](#)

### [Torna all'inizio](#)

## Possibile causa: l'orologio del computer non è sincronizzato

Se utilizzi credenziali valide, l'orologio potrebbe non essere sincronizzato. Su Linux o macOS, esegui `date` per controllare l'ora.

```
$ date
```

Se l'orologio di sistema non viene corretto nel giro di pochi minuti, utilizza `ntpd` per sincronizzarlo.

```
$ sudo service ntpd stop
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

In Windows utilizza le opzioni di data e ora nel pannello di controllo per configurare l'orologio di sistema.

[Torna all'inizio](#)

## La firma non corrisponde agli errori

Esempio di testo di errore:

```
$ 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.
```

Quando AWS CLI esegue un comando, invia una richiesta crittografata ai AWS server per eseguire le operazioni di AWS servizio appropriate. Le credenziali (la chiave di accesso e la chiave segreta) sono coinvolte nella crittografia e AWS consentono l'autenticazione della persona che effettua la richiesta. Ci sono diversi aspetti che possono interferire con il corretto funzionamento di questo processo, come illustrato di seguito:

Possibile causa: l'orologio non è sincronizzato con i server AWS

Per proteggersi dagli [attacchi di tipo 'replay' \(riproduzione\)](#), puoi utilizzare l'ora corrente durante il processo di crittografia/decrittografia. Se l'ora del client e quella del server non corrispondono per un intervallo superiore a quello consentito, il processo può avere esito negativo e la richiesta viene rifiutata. Ciò si può verificare anche quando esegui un comando in una macchina virtuale il cui orologio non è sincronizzato con l'orologio del computer host. Una possibile causa è quando la macchina virtuale va in letargo e dopo il risveglio impiega del tempo per sincronizzare l'orologio con la macchina host.

Su Linux o macOS, esegui `date` per controllare l'ora.

```
$ date
```

Se l'orologio di sistema non viene corretto nel giro di pochi minuti, utilizza `ntpd` per sincronizzarlo.

```
$ sudo service ntpd stop
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

In Windows utilizza le opzioni di data e ora nel pannello di controllo per configurare l'orologio di sistema.

### [Torna all'inizio](#)

Possibile causa: il sistema operativo sta gestendo male AWS le chiavi che contengono determinati caratteri speciali

Se le AWS chiavi includono determinati caratteri speciali, ad esempio `-`, `,` o `+ /%`, alcune varianti del sistema operativo elaborano la stringa in modo errato e causano un'interpretazione errata della stringa chiave.

Se elaborate le chiavi utilizzando altri strumenti o script, ad esempio strumenti che creano il file delle credenziali su una nuova istanza durante la sua creazione, tali strumenti e script potrebbero avere una propria gestione dei caratteri speciali che li trasforma in qualcosa che non è più riconoscibile. AWS

Ti consigliamo di rigenerare la chiave segreta per ottenerne una che non includa il carattere speciale che causa problemi.

### [Torna all'inizio](#)

## Nessun errore rilevato dalla console Windows

Testo di errore di esempio:

```
$ aws s3 ls
No Windows console found. Are you running cmd.exe?
```

Quando si utilizza un AWS CLI comando, viene visualizzato il messaggio «Nessuna console Windows trovata. Stai eseguendo cmd.exe?» messaggio di errore. Questo di solito è un errore

visualizzato per la AWS CLI versione 1 se il Python `prompt_toolkit` che hai installato non è aggiornato. Per risolvere questo problema, installa una versione recente di `prompt_toolkit` sul sito Web di [Python](#).

[Torna all'inizio](#)

## SSLerrori nei certificati

Possibile causa: AWS CLI non si fida del certificato del proxy

Testo di errore di esempio:

```
$ aws s3 ls
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed
```

Quando si utilizza un AWS CLI comando, viene visualizzato un messaggio `[SSL : CERTIFICATE_VERIFY_FAILED] certificate verify failed` di errore. Ciò è dovuto al fatto che il certificato del proxy AWS CLI non è considerato attendibile a causa di fattori quali l'autofirma del certificato del proxy e l'autorità di certificazione (CA) della società è stata scelta come Autorità di certificazione (CA). In questo modo non è AWS CLI possibile trovare il certificato principale CA dell'azienda nel registro CA locale.

Per risolvere questo problema, indica AWS CLI dove trovare il file aziendale utilizzando l'impostazione del `.pem` file di [ca\\_bundle](#) configurazione, l'opzione della riga di [--ca-bundle](#) comando o la variabile di [AWS\\_CA\\_BUNDLE](#) ambiente.

[Torna all'inizio](#)

Possibile causa: la configurazione non punta alla posizione corretta del certificato radice CA

Esempio di testo di errore:

```
$ aws s3 ls
SSL validation failed for regionname [Errno 2] No such file or directory
```

Ciò è causato dal fatto che la posizione del file del bundle dell'Autorità di certificazione (CA) non è configurata correttamente in. AWS CLI Per risolvere questo problema, confermate dove si trova il `.pem` file aziendale e aggiorna la AWS CLI configurazione utilizzando l'impostazione del file di [ca\\_bundle](#) configurazione, l'opzione della riga di [--ca-bundle](#) comando o la variabile di [AWS\\_CA\\_BUNDLE](#) ambiente.



[Torna all'inizio](#)

Possibile causa: la tua configurazione non utilizza la versione corretta Regione AWS

Esempio di testo di errore:

```
$ aws s3 ls
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed
```

Potrebbero verificarsi errori o risultati imprevisti se uno non Servizio AWS è disponibile per i dati specificati Regione AWS o se le risorse si trovano in un altro Regione AWS. Per la risoluzione dei problemi, consulta [the section called “Verifica che il Regione AWS tuo AWS CLI comando stia usando”](#).

[Torna all'inizio](#)

Possibile causa: la tua TLS versione deve essere aggiornata

Esempio di testo di errore:

```
$ aws s3 ls
[SSL: UNSAFE_LEGACY_RENEGOTIATION_DISABLED] unsafe legacy renegotiation disabled
```

Servizio AWS Sta utilizzando una versione TLS incompatibile con la TLS versione del tuo dispositivo. Per risolvere questo problema, esegui l'aggiornamento a una TLS versione supportata. Per ulteriori informazioni, consulta [the section called “Applicazione di una versione minima TLS”](#).

[Torna all'inizio](#)

## Errori non validi JSON

Esempio di testo di errore:

```
$ 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}
```

Quando si utilizza un AWS CLI comando, viene visualizzato un messaggio di errore `Invalid JSON` ». Di solito si tratta di un errore visualizzato quando si immette un comando con un JSON formato previsto e AWS CLI non è possibile leggerlo JSON correttamente.

Possibile causa: non hai inserito un valore valido JSON AWS CLI per l'uso

Conferma di aver JSON inserito un valore valido per il tuo comando. Ti consigliamo di utilizzare un JSON validatore JSON se hai problemi di formattazione.

Per un JSON utilizzo più avanzato nella riga di comando, prendi in considerazione l'utilizzo di un JSON processore da riga di comando, ad esempio `jq`, per creare JSON stringhe. Per maggiori informazioni su `jq`, consulta il [repository jq](#) su GitHub

### [Torna all'inizio](#)

Possibile causa: le regole di quotazione del terminale impediscono l'invio di JSON messaggi validi a AWS CLI

Prima di AWS CLI ricevere qualcosa da un comando, il terminale elabora il comando utilizzando le proprie regole di citazione ed escape. A causa delle regole di formattazione di un terminale, alcuni JSON contenuti potrebbero essere rimossi prima che il comando venga passato a AWS CLI. Quando formuli i comandi, assicurati di utilizzare le regole di quotazione [del tuo terminale](#).

Per risolvere i problemi, usa il `echo` comando per vedere come la shell gestisce i tuoi parametri:

```
$ echo {"ReadCapacityUnits":15,"WriteCapacityUnits":10}
ReadCapacityUnits:15 WriteCapacityUnits:10
```

```
$ echo '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}'
{"ReadCapacityUnits":15,"WriteCapacityUnits":10}
```

Modifica il comando finché non viene restituito il comando `until validJSON`.

Per una risoluzione dei problemi più approfondita, usa il `--debug` parametro per visualizzare i log di debug in quanto mostreranno esattamente ciò che è stato passato a AWS CLI

```
$ 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']
```

Usa le regole di quotazione del tuo terminale per risolvere eventuali problemi riscontrati dai tuoi JSON input quando vengono inviati a AWS CLI. Per ulteriori informazioni sulle regole di quotazione, vedere [the section called “Citazioni con stringhe”](#)

**Note**

Se hai problemi JSON a ottenere la validità su AWS CLI, ti consigliamo di aggirare le regole di quotazione di un terminale per l'immissione JSON dei dati utilizzando Blobs per passare i dati JSON direttamente a AWS CLI. Per ulteriori informazioni su Blobs, consulta [Blob](#)

[Torna all'inizio](#)

## Risorse aggiuntive

Per ulteriore assistenza con i tuoi AWS CLI problemi, visita la [AWS CLI community](#) su GitHub o la [AWS re:Post community](#).

[Torna all'inizio](#)

# Cronologia dei documenti della Guida per l'utente di AWS CLI

La tabella seguente descrive importanti aggiunte alla Guida per l'AWS Command Line Interface utente, a partire da gennaio 2019. Per ricevere notifiche sugli aggiornamenti della documentazione, puoi sottoscrivere il feed RSS.

Modifica	Descrizione	Data
<a href="#">Informazioni aggiornate sulle credenziali e sull'autenticazione.</a>	Istruzioni ed esempi aggiornati sulle credenziali e sui metodi di autenticazione. Ciò include l'aggiornamento delle pagine di configurazione pertinenti. Per far fronte a questo aumento della documentazione, gli argomenti pertinenti sulle credenziali sono stati spostati nella nuova sezione <a href="#">Autenticazione e credenziali di accesso</a> .	31 marzo 2023
<a href="#">I contenuti per la AWS CLI V1 e la V2 sono ora separati nelle rispettive guide</a>	Per motivi di chiarezza e facilità, i contenuti della AWS CLI versione 1 e della AWS CLI versione 2 sono ora separati nelle rispettive guide. Per la AWS CLI versione 2, consulta la <a href="#">Guida per AWS Command Line Interface l'utente</a> più recente	2 novembre 2021
<a href="#">Sono state aggiunte AWS CLI informazioni sull'alias</a>	Sono state aggiunte AWS CLI informazioni sull'alias. Gli alias sono scorciatoie che è possibile creare in AWS	11 marzo 2021

---

	Command Line Interface (AWS CLI) per abbreviare i comandi o gli script utilizzati di frequente.	
<a href="#">Informazioni aggiornate sull'output del filtro</a>	Informazioni aggiornate per i filtri e spostate nella relativa pagina.	1° febbraio 2021
<a href="#">Annuncio di deprecazione per Python 2.7, 3.4 e 3.5</a>	Python 2.7 è stato dichiarato obsoleto dalla Python Software Foundation il 1° gennaio 2020. In futuro, i clienti che utilizzano la AWS CLI versione 1 dovrebbero passare all'utilizzo di Python 3, con un minimo di Python 3.6. Il supporto per Python 2.7 è obsoleto per le nuove versioni della versione 1 a partire dal 19/07/2021. AWS CLI Python 3.4 e 3.5 sono obsoleti a partire dal 01/02/2021.	29 gennaio 2021
<a href="#">È stato aggiunto un esempio di script di Amazon S3</a>	È stato aggiunto un esempio di script del ciclo di vita di Amazon S3.	15 ottobre 2020
<a href="#">Aggiunto un esempio di script di Amazon EC2</a>	È stato aggiunto un esempio di script di tipo di istanza Amazon EC2.	15 ottobre 2020
<a href="#">Sono state aggiunte informazioni sui tentativi</a>	È stata aggiunta una pagina dei nuovi tentativi per le caratteristiche e il comportamento dei nuovi tentativi in AWS CLI	17 settembre 2020

---

<a href="#">Pagina di impaginazione lato server e lato client</a>	Informazioni di impaginazione aggiornate e centralizzate su un'unica pagina.	17 agosto 2020
<a href="#">Pagina dei comandi s3 aggiornata</a>	Aggiornata la pagina dei comandi s3 di alto livello con nuovi esempi e risorse.	30 luglio 2020
<a href="#">Informazioni di installazione aggiornate</a>	Le informazioni di installazione, aggiornamento e disinstallazione per Linux, macOS e Windows sono state aggiornate.	19 maggio 2020
<a href="#">Aggiornato per rimuovere il supporto per Python 2.6 e 3.3 dalla AWS CLI versione 1</a>	A partire dal 10 gennaio 2020, la AWS CLI versione 1 non supporta più l'utilizzo delle versioni 2.6 o 3.3 di Python. È necessario eseguire l'aggiornamento a una versione più recente di Python per AWS CLI utilizzare la versione 1.17 o successiva.	10 gennaio 2020
<a href="#">Nuova sezione MFA</a>	Aggiunta una nuova sezione che descrive come accedere alla CLI utilizzando i ruoli e l'autenticazione a più fattori.	3 maggio 2019
<a href="#">Aggiornamento alla sezione «Utilizzo della CLI»</a>	Miglioramenti e aggiunte importanti alle istruzioni e alle procedure di utilizzo.	7 marzo 2019
<a href="#">Aggiornamento alla sezione «Installazione della CLI»</a>	Principali miglioramenti e aggiunte alle istruzioni e alle AWS CLI procedure di installazione.	7 marzo 2019

[Aggiornamento alla sezione  
«Configurazione della CLI»](#)

Principali miglioramenti e  
aggiunte alle istruzioni e  
alle procedure AWS CLI di  
configurazione.

7 marzo 2019